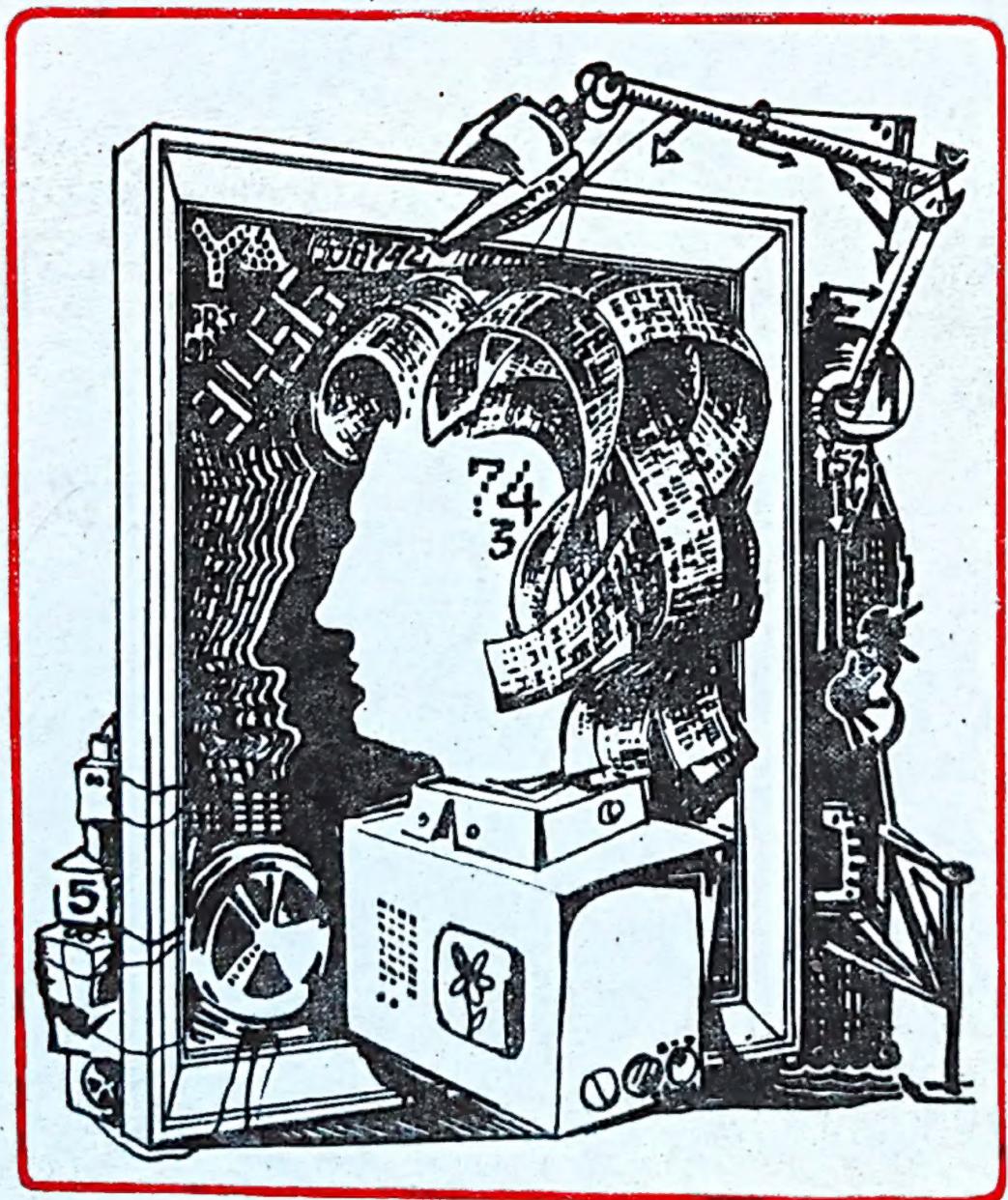


Stelian Guțu • Liviu Dumitrașcu

Elemente de **INTELIGENȚĂ ARTIFICIALĂ** pentru conducerea operativă a producției



STELIAN GUTU și LIVIU DUMITRĂȘCU

ELÈMENTE DE INTELIGENȚĂ
ARTIFICIALĂ
PENTRU
CONDUCÈREA OPERATIVĂ
A PRODUCȚIEI

EDITURA ACADEMIEI REPUBLICII SOCIALISTE ROMÂNIA
București, 1983

P R E F A T Ă

Trăsătură esențială a naturii umane, factor hotărîtor în evoluția vieții pe pămînt, inteligența continuă să rămînă și astăzi, în mare parte, o enigmă. Omul dispune în prezent de unelte extrem de perfecționate, capabile să lucreze singure, precis și economic, după programe inteligente create de el. Există și mașini — calculatoarele electronice — în stare să preia unele funcții ale gîndirii bazate pe memorie și calcul. Sînt însă acestea, oare, inteligente? Greu de dat un răspuns precis la această întrebare. Cert este însă că, la ora actuală, un mare număr de cercetători din lumea întreagă sănătă preocupați să extindă inteligența dincolo de granița umană, să facă în așa fel încît unele obiecte, mecanisme, să acționeze intelligent. Autorii lucrării de față s-au străduit să prezinte în cuprinsul volumului de față cîteva idei și realizări originale proprii în domeniul amintit, avînd în vedere cazul particular al conducerii operative a producției.

Pornind de la necesitatea de a exprima gîndirea — și deci de a valorifica inteligența — prin limbaj, autori propun un instrument, pe care îl vor folosi de altfel pe tot parcursul lucrării, numit: model de reprezentare și manipulare a cunoștințelor economice INTELEC (INTELIGENȚĂ ECONOMICĂ). Acesta, dotat cu un limbaj simplificat față de limbajul natural, are avantajul de a putea fi realizat cu ușurință pe calculatorul electronic.

Suportul informatic al modelului INTELEC este o bază de date relațională (BDR), dotată cu o structură flexibilă care permite adăugarea de noi caracteristici sau ștergerea lor fără a afecta cu nimic informația deja existentă în bază. Modul de utilizare al BDR prin intermediul limbajului INTELEC este ilustrat printr-o serie de exemple.

Completarea informațiilor de detaliu se face pornind de la o frază cu conținut mai mare de generalitate care se folosește drept „prototip” generator. Există și posibilitatea inversă, de sintetizare a unor informații de detaliu. Regăsirea se face fie direct, prin combinarea cuvintelor „cheie” caracteristice, fie cu ajutorul unor procedee indirekte, cum ar fi, de exemplu, utilizînd predicatele: [ESTE OAZ PARTICULAR AL], [ESTE (ÎN ACEST CAZ)], [SE COMPUNE DIN (ANSAMBLU)], [SE COMPUNE DIN (SUCCEZIUNE)] etc., și prelucrarea cărelor relationale (vezi capitolul 3 al lucrării).

Modelul și limbajul INTELEC alcătuiesc un instrument practic adaptabil nevoilor conducerii operative a producției din întreprinderi. Avînd la bază conceptul de conducere operativă promovat de o serie de institute de învățămînt superior și institute de cercetare din România, concept ale cărui elemente fundamentale se axează pe folosirea la maximum a produselor informaticice — pachete de programe — în activitatea de planificare-progra-

mare-lansare-urmărire și control a producției privită ca un tot unitar, autorii propun un mod original de participare a calculatorului la realizarea acestui concept, mod care poate fi descris pe scurt astfel:

Să considerăm următoarele situații în care se poate afla, într-o întreprindere, un calculator electronic destinat conducerii operative a producției: (a) calculatorul în postură de consilier pentru problemele de conducere operativă a producției; (b) calculatorul-executant al unor programe de optimizare a producției și (c) calculatorul-robot, rezultat prin combinarea celor două proprietăți sus amintite, la care se adaugă dependența și acțiunea complexă (directă sau indirectă), reciprocă, a calculatorului față de mediul înconjurător. Modelul, capabil să treacă prin cele trei faze de mai sus, de la simplu la complex, asigurând preluarea treptată de către calculator a unor funcții ale gândirii și activității umane în domeniul conducerii operative a producției, a fost numit de autori model intelligent de conducere operativă a producției. De remarcat faptul că rolul omului în dirigarea și supravegherea proceselor efectuate de calculator nu dispare complet nici chiar în stadiul cel mai avansat de automatizare; Ponderea acestor intervenții este însă diferită și scade pe măsura preluării de către calculator a unor activități inteligențe (vezi capitolul 5 al lucrării).

Pe baza conceptului de model intelligent și cu ajutorul instrumentului INTELEC, s-a abordat un exemplu concret de conducere operativă a producției în ramura construcțiilor de mașini. Pentru producția de serie mică și unică, există un pachet de programe, construit și experimentat de unul din autori (S. Guțu) în cîteva întreprinderi din țară, pachet care s-a dovedit deosebit de suplu în planificarea și programarea producției organizate pe posturi de lucru, atât la nivel global (anual, trimestrial), cît și pe perioade mai scurte (luni, decade, săptămâni). Aceasta oferea conducerii, cu mijloace relativ modeste (fără echipamente automate de urmărire a producției), posibilitatea de a cunoaște în orice moment, cu o precizie suficientă, ce anume poate și ce nu realizează din ceea ce și-a propus, sau are de gînd să-și propună, ca program de fabricație.

Integrat într-un concept mai larg de conducere operativă pentru întreprinderile constructoare de mașini, pachetul de programe SOFTPLAN (a cărui descriere este dată în cadrul capitolului 6 al lucrării) verifică, în ultima instanță, veridicitatea tuturor programelor de fabricație atât la nivel anual (pentru stabilirea contractelor și definitivarea planurilor de fabricație, aprovizionare, colaborare etc.), cît și la nivelul fiecărei perioade operative în parte (lună, decadă etc.), cu care ocazie are în vedere recuperarea restanțelor și întîrzierilor, precum și noile priorități apărute pe parcurs. Modelul de mai sus, care oferă răspuns la majoritatea întrebărilor legate de conducerea producției în întreprinderile constructoare de mașini organizate pe principiul posturilor de lucru, a fost transpus în limbaj INTELEC. În acest fel, oricine dorește să știe ce are de făcut într-o anume etapă, ce sarcini prioritare are de rezolvat, cu ce greutăți va fi confruntat etc., se poate adresa calculatorului care, pe post de consilier, răspunde prompt dînd indicațiile necesare. De exemplu, celor care doresc să experimenteze pachetul de programe SOFTPLAN, calculatorul le poate furniza indicații detaliate despre ce anume trebuie să facă, ce trebuie să urmărească, ce modul de program trebuie să aleagă pentru pasul următor în funcție de rezultatele obținute etc. Literatura de specialitate clusează acest gen de aplicații în categoria programelor de inteligență artificială de tip „expert”. Cuplul „forță-inteligentă” este realizat, în cazul

de față, pe de-o parte de optimizarea încărcării capacitaților de producție efectuată de SOFTPLAN, iar pe de altă parte de modelul de acțiune generalizat: **〈PROGRAM PRINCIPAL〉 — 〈SUBPROGRAM〉 — 〈SALVARE〉 — 〈DECIZIE〉 — 〈ÎNVĂȚARE〉** (descriși în capitolul 4 al lucrării), pe baza căruia a fost construit modelul de conducere operativă a producției pentru întreprinderile constructoare de mașini. De remarcat faptul că, în etapa actuală, cele două elemente: **forță** (pachet de programe) și **inteligentă** (răspunsuri la întrebările utilizatorului), acționează separat, legătura fiind realizată de om care comunică modelului rezultatele obținute prin aplicarea pachetului de programe și așteaptă indicații de la calculator. Cuplarea elementului „**forță**” cu acela de „**inteligentă**” — posibilă într-o etapă viitoare — ar duce la transformarea modelului într-un automat complex de tip robot, capabil să aleagă singur variantele corespunzătoare obiectivelor întreprinderii și să urmărească realizarea lor.

Modelul de conducere operativă a producției descris în această lucrare nu este numai un model teoretic. Se dau indicații precise (vezi capitolul 7 al lucrării) în ceea ce privește posibilitatea de implementare a modelului INTELEC și a bazei sale de date relaționale. Un exemplu de creare și exploatare a structurii modelului INTELEC, cu ajutorul unui limbaj funcțional realizat de autori și denumit sistem de bază de date cu structură generalizată (SBDSG), demonstrează viabilitatea soluției propuse. În ceea ce privește partea de „**forță**”, pachetul de programe SOFTPLAN a fost experimentat pînă acum în cîteva întreprinderi, parte din rezultatele obținute fiind rezumate și în această lucrare (vezi capitolul 6).

În încheiere se dau indicații referitoare la realizarea analizoarelor semantic și sintactic ale limbajului INTELEC, o posibilitate fiind și aceea a folosirii limbajului LISP pentru care se dau și unele exemple concrete de programe.

*

Lucrarea, în ansamblul ei, este o colecție de idei și rezolvări inedite care iese din tiparele obișnuite ale genului.

După un capitol introductiv în domeniul inteligenței în general, cu referiri la inteligența artificială și un al doilea de definiții și axiome referitoare la decizii, se trece direct la miezul problemei: se definește din punct de vedere matematic modelul INTELEC, se construiește baza de date relațională și se dau exemple de utilizare a lor în conducerea producției.

În continuare, capitolul următor se ocupă de o problemă specială de inteligență artificială: **planificarea comportamentului** — cu aplicații directe în conducerea operativă a producției. Aici se definește mecanismul care stă la baza părții aplicative și anume modelul de comportare intelligentă în orice situație de conducere operativă a producției. Urmează definirea conceptului de conducere operativă, descrierea pachetului de programe SOFTPLAN și, în final, în ultimul capitol, se face legarea tuturor acestor elemente în cadrul modelului de conducere operativă a producției în ramura construcțiilor de mașini.

Deși axată pe o aplicație concretă, lucrarea rămîne deschisă și altor genuri de utilizări ale modelului și limbajului INTELEC în domeniul inteligenței artificiale. Din acest punct de vedere autorii rămîn datori și speră în continuarea și dezvoltarea prezentului volum. La actuala apariție, lucrarea întrunește toate condițiile pentru a stîrni interesul și a îndemna cititorii la numeroase reflexii și aprofundări. De un real folos — sperăm — cercetătorilor în domeniu, lucrarea poate fi consultată de un public larg — elevi,

studienți, cadre didactice, ingineri, cadre de conducere din întreprinderi — precum și de toți aceia care doresc să obțină, într-un limbaj accesibil, informații referitoare la ce se poate și mai ales cum se poate aborda și rezolva concret o problemă de conducere a producției cu ajutorul calculatorului pe baza unor concepte și teorii în plină evoluție și perfecționare astăzi în lume.

★

Profităm de acest prilej pentru a exprima încă odată sentimentele noastre de înaltă stima și gratitudine tovarășului profesor dr. docent Mihai Drăgănescu, membru corespondent al Academiei R.S. România, initiatorul și conducătorul mișcării de inteligență artificială românești — actualmente în plină dezvoltare — pentru încurajările și aprecierile adresate cu bunăvoie autorilor acestei lucrări, pentru grija și spiritul de răspundere cu care a avizat tipărirea volumului de față.

De asemenea, aducem pe această cale mulțumirile noastre sincere tovarășului ing. Ioan Georgescu, șeful colectivului de inteligență artificială din cadrul Institutului Central pentru Conducere și Informatică, pentru răbdarea de a fi citit cu atenție manuscrisul și a fi contribuit prin corecțiile și sugestiile sale de înaltă competență și probitate profesională la ridicarea nivelului calitativ al lucrării.

Nu putem încheia aceste rânduri fără a ne exprima recunoștința pentru aportul deosebit adus de către Editura Academiei la tipărirea volumului de față.

AUTORII

C U P R I N S

CAPITOLUL 1. Elemente definitorii ale inteligenței umane în cadrul activităților economico-productive

1.1. Inteligența și mecanismul găndirii	12
1.2. Oportunitatea introducerii unor elemente de inteligență artificială în conducerea operativă a producției	22

CAPITOLUL 2. Modele ale procesului de decizie

2.1. Considerații preliminare	26
2.2. Formalizarea și studiul matematic al deciziei	26
2.3. Dinamica procesului	27
2.4. Decizii cu caracter de rutină	28
2.5. Programarea actului decizional	30
2.6. Conceptul de utilitate. Metoda ELECTRE aplicată pe terminal	31
2.7. Conceptul de multiraționalitate	35

CAPITOLUL 3. Modelul relațional INTELEC, de reprezentare și manipulare a cunoștințelor. Definiție, structură, realizare informatică și mod de utilizare

3.1. Generalități	38
3.2. Definiția limbajului INTELEC de reprezentare și manipulare a cunoștințelor	42
3.3. Predicale, funcții predicative și teoreme	46
3.4. Definiția bazei de date relaționale (BDR)	55
3.5. Principii de utilizare a bazei de date relaționale	57
3.6. Modul de utilizare a bazei de date relaționale	58

CAPITOLUL 4. Planificarea comportamentului în cadrul modelului INTELEC

4.1. Generalități	78
4.2. Planificarea comportamentului în cazul utilizării optime a unui pachet de programe pentru conducerea operativă a producției	78
4.3. Aplicație în cazul utilizării pachetului de programe SOFTPLAN în întreprinderi	82
4.4. Introducerea unor elemente de inteligență artificială în planificarea deciziilor privind alegerea fazelor de optimizare	89

CAPITOLUL 5. Conceptul de model intelligent în conducerea operativă a producției

5.1. Structura conducerii operative a producției	96
5.2. Conceptul de model intelligent	100
5.3. Aplicarea conceptului de model intelligent în conducerea operativă a producției	102

CAPITOLUL 6. Sistemul SOFTPLAN — element de bază în realizarea unui model intelligent pentru conducerea operativă a producției în ramura construcțiilor de mașini	
6.1. Utilitatea sistemului	114
6.2. Modelul matematic	115
6.3. Arhitectura sistemului	128
6.4. Unele rezultate și concluzii obținute prin aplicarea sistemului SOFTPLAN în întreprinderi	131
CAPITOLUL 7. Conducerea operativă a producției cu ajutorul modelului «INTELEC»	
7.1. Obiective	154
7.2. Calculatorul-consilier	154
7.3. Calculatorul-executant	164
7.4. Calculatorul-robot	165
7.5. Realizarea modelului INTELEC	167
Anexă	187
Bibliografie	203
Indice de termeni	207
Abstract	213
Contents	217

CAPITOLUL

1

Elemente definitorii ale inteligenței umane în cadrul activităților economico-productive

Ce este inteligența? Cum funcționează mecanismul gîndirii? Iată întrebări care au frâmînat mintea omenească încă din cele mai vechi timpuri. Filozofia antică excezează prin căutări în acest domeniu. Mai tîrziu, în evul mediu și în perioada renașterii ideile se complică și se diversifică. Dezvoltarea logicii matematice pare să dea un nou impuls cercetărilor legate de rațiune, de inteligență. Tehnica manipulării simbolurilor „obiectivează” cercetările sau este numai o sistematizare a proprietății noastre intuiției? Ce aduc nou psihologia, biologia și neurocibernetica în știința despre gîndire? Și, în sfîrșit, despre așa-zisa Inteligență Artificială (I.A.). Cum poate contribui calculatorul electronic la descoperirea legilor gîndirii și, mai ales la rezolvarea unor probleme complexe, de ordin practic, din activitatea economico-productivă? O serie de observații referitoare la acest subiect sunt prezentate la sfîrșitul capitolului din care decurge oportunitatea investigațiilor în domeniul conducerii operative a producției și demersul logic al întregii lucrări.

1.1. Inteligența și mecanismul gîndirii

A încerca să definești inteligența umană în condițiile în care progresul tehnico-științific de astăzi aduce din zi în zi probe tot mai evidente în favoarea incomensurabilității gîndirii omenesti înseamnă, într-o oarecare privință, a tinde să cuprinzi infinitul. Și, totuși, încă din cele mai vechi timpuri, oamenii au fost tentați să se autoanalyzeze, să se autodefinească în ceea ce îi deosebea esențial de restul ființelor care populau globul terestru. Celebra deviză „Cunoaște-te pe tine însuți” gravată pe frontispiciul templului de la Delfi a constituit unul din principiile de bază ale filozofiei antice grecești. La Anaxagora, de exemplu, *Inteligență* înseamnă *Ordine*; lumea este ordonată și această ordine nu este decât expresia unei inteligențe independentă de ea. O „inteligență superioară” (*Nous-ul*) produce ordine, iar lumea reală, ordonată, este inteligibilă, logică și exprimabilă ([28]) p. 88).

Socrate acordă o importanță primordială *conceptului* pe baza căruia definește știință (la el, arta de a defini): „Numai conceptul poate pune de acord omul cu el însuși și cu ceilalți” spunea el ([79] p. 105). Aristotel atribuie lui Socrate două mari descoperiri: „procedeul inducției și definiția generală, principii care, amindouă, constituie începutul orientării științe”. Este interesant de amintit aici procedeul metodologic al lui Socrate, socotit și astăzi o cale inteligentă de stabilire a adevărului.

Prima etapă a discursului socratic (numită de el *ironia*), avea drept scop de a strînge adversarul în cușca propriilor sale contradicții. Ea constă în investigația inteligentă, în arta de a pune întrebări în așa fel încât răspunsurile firești și logice ale interlocutorului să nu trezească suspiciuni din partea acestuia și să servească pe de altă parte demonstrației din etapa următoare.

A doua etapă (numită de el *maieutica*) era momentul constituirii adevărului pe baza analizei răspunsurilor primite și generalizarea unor aspecte particulare care duceau în cele din urmă la definirea conceptului.

Cel care pune însă bazele „gîndirii despre gîndire” descoperindu-i legile, constituind-o definitiv ca știință a științelor este, fără îndoială, Aristotel. Cea mai mare descoperire a acestuia o constituie *actul de auto-reflexie al gîndirii*, actul cel mai înalt al inteligenței prin care se pune în evidență însăși esența realității. Aristotel face o distincție netă între *formă* și *materie*. Cunoașterea realității începe prin contactul cu materia prin intermediul simțurilor. Gîndirea extrage apoi esențialul — *formele inteligibile* — care devin obiect de prelucrare la nivelul superior al intelectului.

Această teorie seamănă uimitor de mult cu ceea ce ne oferă astăzi biologia și neurocibernetica referitor la funcționarea creierului omenesc. Se afirmă că „una din condițiile esențiale ale gîndirii raționale este clasicarea senzațiilor” și că „operația de clasificare este o operație creatoare a rețelei neuronale” ([63] p. 275), ori clasificarea se realizează cu ajutorul conceptelor — a „formelor” în limbaj aristotelic. Să vedem mai departe care sunt operațiile intelectuale prin care gîndirea ajunge de la sen-



zație la formele sale „pure” — intelibile. Aristotel menționează, în primul rînd *imaginea* care supune materialul sensibil unei prime prelucrări, degajind calități generale și obținind imagini. „Sufletul intelectiv (practic) gîndește, deci, formele prin imagini și, în măsura în care în acestea îi este determinat obiectul de urmărit sau de evitat, în aceeași măsură, chiar fără senzație, cînd se află în fața imaginilor, se pune în mișcare” ([4] p. 96).

Și aici este interesant de menționat rolul important pe care se pare că îl au retina și centrii nervoși subcorticali în formarea imaginilor, în extragerea unor caracteristici esențiale ale acestora și în efectuarea chiar a unor prime prelucrări de informații înainte de prelucrarea finală în contextul cerebral. De altfel, unele încercări moderne de a imita pe cale artificială aceste operații utilizînd mecanisme complicate de recunoaștere a formelor, pe baza extragerii unor caracteristici esențiale ale obiectelor și a comparării lor cu caracteristici similare prememorate, au dus la realizarea cu succes a unor funcții de bază pentru roboți.

Aristotel consideră că gîndirea pleacă de la imagini, extrage ideile intelibile din lucrurile sensibile, operația ultimă fiind efectuată de intelect care are intuiția directă a esențelor, a formelor intelibile ([4] p. 97–98). Dacă admitem, conform teoriei genetice moderne, că aceste „esențe” ar putea fi informații ereditare pe care individul le primește îndăr de la părinții săi și care reflectă de fapt evoluția întregii specii umane și chiar mai mult, a materiei pe planeta noastră, explicația aristotelică a procesului gîndirii ne apare cu totul plauzibilă. Esențele prin care se exprimă proprietățile cele mai generale ale lucrurilor sunt indicate de Aristotel sub forma celebrelor sale *categorii* ([5] p. 119–191) pe care le redăm: (1) esența (arată ce este un lueru); (2) cantitatea (cît de mare); (3) calitatea (cum este alcătuit); (4) relația (în ce relație); (5) locul (unde); (6) timpul (cînd); (7) situația (în ce situație); (8) posesia (ce are); (9) acțiunea (ce face); (10) pasiunea (ce suferă).

Să privim o clipă acest tabel și să ne amintim că celebra teorie lingvistică a lui Fillmore asupra cazurilor „profunde” [31], în care fiecare concept este explicat prin „scene” în care rolurile sunt jucate de alte concepte iar acestea la rîndul lor de altele ș.a.m.d. pînă la nivelul cel mai înalt, trebuie să pornească în fond de la niște concepte foarte generale, de tipul celor găsite și enunțate încă acum 2000 de ani în urmă de Aristotel, De altfel, modelul relațional dezvoltat în prezenta lucrare (capitolul 3) utilizează și o parte din noțiunile cu caracter general enumerate mai sus.

Pe lîngă „genurile supreme” pe care le-am văzut, Aristotel introduce alte cinci categorii speciale, numite de el „predicabile”. Acestea sunt:

1. *Genul* — conceptul avînd cea mai mare generalitate, cuprinzînd un număr de specii.
2. *Specia* — determinată în interiorul unui gen, cuprinzînd toate caracterele genului (genul proxim) la care se adaugă particularitățile grupului pe care îl reprezintă.
3. *Diferența* — caracterele distinctive (diferență specifică).
4. *Propriul* — ceea ce nu poate exista fără esența speciei, fără a fi totuși un element esențial.

5. *Accidentul* – ceea ce se adaugă esenței și poate exista și la alte specii, în cazul analizat apărind cu totul intimplător.

Cele de mai sus ne ajută să înțelegem noțiunile de *definiție* (care exprimă esența lucurilor), *judecată* (raportul dintre general și individual), *silogism* (metodologia demonstrației), *ipoteză* (premisă), *demonstrație* și *raționament*. De asemenea, Aristotel definește *dialectica*, *inducția* și se ocupă de raționamentele greșite (sofisme) pe care le analizează și le clasifică din punct de vedere metodologic ([6] p. 268–377).

Putem spune deci, cu pretenția de a nu greși prea mult, că nici una din problemele actuale ale științei despre gindire nu a fost ocolită de Aristotel și că marea majoritate a conceptelor fundamentale descoperite de el se regăsește sub diferite aspecte la filozofii și logicienii din evul mediu și la cei moderni, ele constituind punctul de plecare al unor teorii despre inteligență dezvoltate de ramurile moderne ale științei.

Astfel, deosebirea între formă și materie este exprimată de școala scolastică medievală prin cele două „intenții”: *intentio prima* (cunoașterea unui obiect în mod direct, prin intermediul simțurilor) și *intentio secunda* (îmaginea formată în gindirea noastră despre obiectul în cauză). Teoria cunoașterii a lui Aristotel este completată de experimentaliști care adaugă „autorității” și „rațiunii” o a treia cale de cunoaștere, esențială după părerea lor, „experiență” (Roger Bacon). Silogismul și logica, în general, nu au nici o valoare dacă nu sunt sprijinite de experiență. „Cea mai bună demonstrație este experiența, dacă ea nu se oprește la experimentul însuși”, spune Francis Bacon ([10] p. 58). Totuși, el nu este capabil să dea pentru actul gindirii decit explicații aparente, puerile, instrumentul (*Organonul*) pe care îl propune neavând decit o valoare pur metodologică.

Logica transcendentală a lui Immanuel Kant, în schimb, are ca obiect însuși intelectul uman pe care filozoful vrea să-l „disece” pentru a deduce astfel puterile rațiunii omenești, ale inteligenței umane. „Intelectul este, în general vorbind, facultatea *cunoașterilor*. Acestea constau în raportul determinat al reprezentărilor date la un obiect. Dar *obiectul* este ceva în al cărui concept este unit diversul unei intuiții date. „Pentru a cunoaște ceva în spațiu, de exemplu o linie, eu trebuie să-o trag și să efectuez astfel sintetic o legătură determinată a diversului dat”, spune Kant ([57], p. 132–133) și, mai departe: „Noi nu putem gîndi nici un obiect fără ajutorul categoriilor; noi nu putem cunoaște nici un obiect gîndit fără ajutorul intuițiilor cărora le corespund aceste concepte” ([57], p. 159). Kant revine astfel la căutarea conceptelor fundamentale ale gindirii, și, la fel ca Aristotel, enunță un tablou sintetic al *categoriilor și judecăților*.

Se regăsește aici unele concepte aristotelice, cum sunt esența, cantitatea, calitatea, relația, în timp ce altele, ca de exemplu situația, posesia, acțiunea, sunt excluse de Kant pentru motivul că aparțin sensibilității și nu intelectului. Oricum, rămîne mai departe valabilă ideea unor categorii abstrakte care să facă posibilă gindirea prin funcția lor unificatoare, dar care, spre

deosebire de concepțele aristotelice, nu reprezintă sinteza unor percepții directe, senzoriale, ci există în intelect *a priori* și nu capătă consistență decit în urma experienței. Fără să o explice, Kant se mulțumește să acorde gîndirii o putere întrinsecă, transcendentală, de a realiza *sinteze* avind două

TABLOUL LUI I. KANT (CONF. [28], p. 647)

Criterii de clasificare	Judecăți	Categorii
După cantitate	Generale Particulare Singulare	Unitatea Pluralitatea Totalitatea
După calitate	Afirmative Negative Nedefinite	Realitatea Negația Limitarea
După relație	Categorice Ipotetice Disjunctive	Substanța și accidentul Cauza și efectul Reciprocitatea dintre activ și pasiv
După modalitate	Problematice Asertorice Apodictice	Posibilitatea și imposibilitatea Existența și non-existența Necesitatea și contingenta

premise fundamentale: *conceptul pur* și *experiența*. Inelul de legătură între *fenomen*, considerat ca formă de manifestare a obiectului în afara intelectului, și *categorie* — formă „vidă” a intelectului, îl constituie, după Kant, *țimpul*. Kant, ca și Aristotel, acordă *imaginei* rolul de a preluca în timp materialul faptic real și a extrage reprezentări pe care să le supună apoi *intuiției*. Imaginea trasează totodată „*a priori*” în intuiția spațiului și timpului anumite cadre pentru fenomene — un fel de tipare care sunt numite de Kant *schemă transcendentală* (schemă ale conceptelor intelectului pur). Astfel, unitatea, pluralitatea și totalitatea au ca tipar în care se regăsesc pe planul fenomenelor cantitative — numărul. Categoriile cauză și efect au ca schemă succesiunea și.m.d. Aceasta constituie, după părerea noastră, o nouă apropiere de teoria modernă care încearcă să explice gîndirea prin *scene* (schemă) în care rolurile sunt jucate de *concepțe*, teorie pe care am mai menționat-o.

Este, de asemenea, interesant de amintit aici și faptul că o serie de modele semiotice ale activității intelectuale a omului, realizate astăzi cu ajutorul calculatorului electronic, au la bază sisteme de concepțe date „*a priori*” și introduse în calculator, ierarhizate și corelate după „schemă” originale ale autorilor. Astfel, modelul lui Pospelov ([66] p. 131—158),

Înțelegând de la teoria genetică, încearcă să reprezinte lumea exterioară subiectului printr-o mulțime de obiecte împreună cu legile lor de interacțiune. Între conexiunile din interiorul sistemului de semne și conexiunile obiectelor lumii reale se creează o anumită concordanță, astfel încit, într-un anume sens, structurile celor două sisteme — real și modelul său — sunt izomorfe. Sistemul de semne astfel creat poartă numele de *sistem de semne de gradul întâi* sau *sistem semiotic pseudofizic*. Acesta din urmă poate fi și el, la rîndul lui, analizat și codificat iar sistemul de semne care servește la reprezentarea calităților lui, a structurii sau a interacțiunii dintre elementele sale, poartă numele de *sistem de semne de gradul al doilea*. Potrivit legii inducției, asemenea construcții pot continua la infinit dinăuntru la sisteme de semne de gradul al treilea, al patrulea și.a.m.d. Se pare că există o asemănare între un asemenea model complex și modul de lucru al gîndirii științifice creațoare. Inteligența umană folosește astfel de sisteme ierarhizate atunci cînd cu ajutorul limbajului natural trece la definirea altor limbaje proprii unor domenii particulare ale științei, artei, vieții sociale etc.

Aristotel și Kant au avut în vedere o anume corelație intrinsecă între obiecte și fenomene, și s-au bazat în teoriile lor pe capacitatea intelectului de a extrage aceste conexiuni spațio-temporale. În modelele semiotice despre care vorbeam, obiectele lumii exterioare sunt de asemenea analizate într-un continuum spațio-temporal rezultînd o serie de *corelații-tip*. În modelul lui Rvaciov ([66] p. 138—139), de exemplu, există cinci predicate de bază cu care se pot exprima legăturile între obiecte în acest continuum :

- obiectul *A* este parte a obiectului *B* (relație de spațiu) ;
- obiectul *A* este comparabil cu obiectul *B* (relație de spațiu) ;
- obiectul *A* este identic cu obiectul *B* (relație de spațiu) ;
- obiectul *A* precede (în timp) obiectul *B* ;
- obiectul *A* este cauza obiectului *B* (obiectul *B* este consecința obiectului *A* și îi urmează în timp acestuia).

În mod analog, modelul lui Pospelov conține 30 de asemenea predicate ([66] p. 141).

Kant vorbea despre proprietatea intelectului de a *intui* soluții, de a crea cadre conceptuale pentru fenomenele observabile ale lumii reale. Cu mult înaintea lui Kant, René Descartes acorda și el intuiției un rol primordial în a ajunge la cunoștința lucrurilor fără nicio temere de eroare, atribuind inteligenței umane două proprietăți fundamentale : *intuiția* și *deducția*. Ce este de fapt această *intuiție* pe care marii filozofi își împrează explicația lor privitor la gîndirea științifică și creativitate?

Este ceea ce numim, în termeni matematici, *capacitatea creierului uman de a construi ipoteze și de a formula concluzii*. Modelele semiotice ale creativității gîndirii încearcă să realizeze această funcție la niveluri ierarhice superioare (sisteme de semne de gradul trei și patru) cu ajutorul unor memorii dinamice în care se înregistrează evoluția în timp a structurilor de obiecte și a unor caracteristici mai generale ale acestora și care sunt supuse apoi diverselor prelucrări statistico-matematice și clasificărilor. Rezultatele sunt apreciate pe baza unor caleule cu cuantificatori imprecisi (de tipul : *există un x avînd proprietatea p, pentru toți x*



avînd proprietatea p , sînt foarte pușini x care au proprietatea p , sînt pușini x care au proprietatea p , aproximativ jumătate din toți x au proprietatea p etc.) ([66] p. 145). Luerul cu asemenea cuantificatori obligă raportarea obiectelor la un univers finit; afirmația unei propoziții p duce la analiza condițiilor în care este valabilă negația ei \bar{p} (non p) și de aici determinarea coordonatelor lui p în universul restrîns considerat. Se simulează astfel la scară restrînsă principiile gîndirii dialectice *teză-antiteză-sinteză*, formulate pentru prima dată de Hegel, principii reluate și dezvoltate apoi pe plan materialist de către clasicii materialismului dialectic Marx, Engels și Lenin.

Dialectica este definită de Engels ca „știință a conexiunii universale” ([1] p. 325) și ca „știință a legilor celor mai generale ale istoriei naturii și societății, ale gîndirii însăși” ([1] p. 367). Apar aici două idei fundamentale: (1) unitatea dintre materie și gîndire, (2) existența unor legi care guvernează mișcarea materiei și a gîndirii. Rezultă că pentru a putea defini inteligența ca proces — și acesta este, de fapt punctul de vedere dialectic — și nu ca ceva dat, existent ca atare, este necesar să cunoaștem și să analizăm aceste legi. Engels le formulează astfel în *Dialectica naturii*: (a) legea trecerii cantității în calitate și invers; (b) legea întreăstrării opușilor; (c) legea negăției.

Toate exemplele care urmează, în intenția de a ne convinge de adevărul obiectiv al acestor legi, sunt luate din fizică, chimie, istorie și biologie. Aceasta este de altfel numită de Engels *dialectica obiectivă* în timp ce *gîndirea dialectică* este numai „reflectarea mișcării prin opuși care, prin conflictul lor permanent și prin trecerea finală a unuia în celălalt, respectiv în forme superioare, determină viața, natura” ([1] p. 509). Atracția și respingerea, polarizarea substanței albuminoide vii și nașterea nucleului celulei, ereditarea și adaptarea, sunt date de el ca exemplu de conflicte care apar permanent în dezvoltare, și cărora nu li se potrivesc schematic categoriile „pozitiv” și „negativ”, cu atât mai mult cu căt „putem concepe ereditatea ca latură pozitivă, conservativă, iar adaptarea ca latură negativă, care distrugă în permanentă caracterele moștenite; dar tot atât de bine putem concepe adaptarea ca activitate creatoare, eficientă, pozitivă iar ereditatea ca activitate contracarantă, pasivă, negativă” ([1] p. 509).

Așadar, reprezentările pe care gîndirea le dă obiectelor nu au valabilitate decât în lumina *conexiunilor* pe care tot ea le stabilește cu lumea înconjurătoare, conexiuni care și ele au o valabilitate temporară fiind în continuă transformare și perfecționare.

V. I. Lenin remarcă la rîndul lui justițea afirmației hegeliene cu privire la dialectica conceptelor: „Obiectul, în afara conceptului, este o simplă denumire; numai în determinările gîndirii și ale conceptului este el ceea ce este” ([2] p. 191). El consideră întreaga dialectică o „teorie a unității contrariilor” ([2] p. 189). Iată cum explică el această noțiune: „*Dialectica*, în general, constă în negarea *primei* teze, în înlocuirea ei cu *a doua* (în trecerea primei teze în a două, în stabilirea legăturii dintre prima și a două etc.). *A doua* poate fi făcută *predicatul primei* — de exemplu finitul este infinit, unul este multiplu, singularul este universal. Nu negația goală, nu negația fără rost, nu negația sceptică, ci oscilarea, îndoiala, sunt caracteristici esențiale în dialectică” ([2] p. 192) — ceea ce ne remîntă de cunoscută exprimare sintetică a principiului îndoielii *Dubito ergo cogito* care constituie „esența metodei de cercetare” așa cum afirma încă în sec. XVII (*1637 — Discours de la méthode*) René Descartes.

Dacă *legile* sint, în dialectica materialistă, comune materiei și gîndirii, *categoriile* nu pot exprima decit adevăra principală a gîndirii și formelor ei, realității. Categoriile, în dialectica materialistă, nu sint determinări absolute, scheme definitive ca la Aristotel și Kant, ci au valabilitate absolută numai într-un domeniu *delimitat în spațiu și în timp*. Ele sint, așa cum le caracterizează Lenin, *momente ale cunoașterii naturii de către om* ([2] p. 167). Nici numărul lor nu este fixat odată pentru totdeauna. Totuși, clasicii marxism-leninismului s-au ocupat în mod special de unele categorii dialectice fundamentale, cum ar fi *singular-particular-general, esență-fenomen, conținut-formă, cauză-efect, necesitate-întîmplare, posibilitate-realitate* și a.

„Cunoașterea este o apropiere permanentă și infinită a gîndirii de obiect” iar „reflectarea naturii în gîndirea omului trebuie înțeleasă în procesul veșnic al mișcării, al apariției contradicțiilor și al rezolvării lor” ([2] p. 165) spune Lenin. Recunoaștem în această afirmație esența dialecticii materialiste. Totuși, despre mecanismul intim al gîndirii, despre ceea ce face și mai ales cum face creierul omenesc pentru a reflecta, mai mult sau mai puțin exact, în continuă perfecționare și antagonism de idei, concepții etc., realitatea, nu aflăm nimic care să ne permită modelarea și reproducerea – cel puțin în mică măsură – a funcțiilor lui, pe cale artificială.

Singura care pare a fi avansat în domeniul unor construcții inteligente artificiale, obținind unele rezultate care, aparent, deși limitate, seamănă în mare măsură prin generalitatea lor cu acelea produse de creierul omenesc în condiții naturale de activitate, este *logica formală* sau, mai exact, *logica matematică* sau *algebra logică*. Despre rezultatele și limitele sale ne vom ocupa în cele ce urmează.

Noțiunea de *sistem formal* corespunde unei perfecționări a metodei axiomatice a lui Ladrrière [59] prin care se descrie în mod explicit organizarea internă a unei științe, punind în evidență toate categoriile de obiecte și propoziții care apar, precum și motivele în virtutea cărora ele sint acceptate. Nu mai avem de-a face, așadar, cu obiecte individuale ci cu concepte; nu se descriu și se analizează fapte ci *relații logice între concepte*. Metoda axiomatică înseamnă trecerea de la știința materială la știință formală. Ceea ce aduce nou și pozitiv axiomatica este eliminarea intuiției, creșterea rigorii științifice, facilitatea corelațiilor și mai ales descooperirea neconcordanțelor din cadrul unui sistem de concepte reprezentate prin simboluri matematice. Totuși, așa cum remarcă Anton Dumitriu, „Metoda axiomatică nu poate face mai mult decât să descrie o știință, să-i arate încheieturile logice, ea nu poate depăși această limită (– nu posedă creativitate – N.N.). Pentru a se putea desfășura, metoda axiomatică are nevoie de cineva *din afara ei*, care gîndește și-i dă impulsul devenirii ei. O știință nu este axiomatizată dinăuntrul ei din afara ei; în ea însăși, fără matematicianul care gîndește și o dirijează nu reprezintă decât o schemă moartă” ([27] p. 12).

Așadar, deși sistemul formal reprezintă gradul suprem de abstracție și de reprezentare a fenomenelor reale și dispune de maximum de obiectivitate în stabilirea relațiilor, el *nu are nimic de-a face cu mecanismul propriu-zis al gîndirii, este numai o simplă coordonare de simboluri*.

Într-adevăr, să luăm spre exemplificare sistemul logico-matematic al lui Whitehead și Russell expus în *Principia mathematica* [28]. Acesta cuprinde: (1) *idei primitive* – propoziții elementare care exprimă ceva dat prin simțuri, fără un anume conținut, ex.: *p, q, r*. Acestea pot lua



două valori : *A* (adevărat) și *F* (fals). Negația, disjuncția, implicația etc. sunt definite în termeni de *p*, *q* și *r*; (2) *axiome* — un număr de propoziții adevărate indiferent de valoarea de adevăr a variabilelor; (3) *reguli de deducție sau de derivare* — reguli care stau la baza demonstrării oricărei teoreme; (4) *teoreme* — care stau la baza calculului propozițional.

Calculul propozițional este singurul sistem formal a cărui necontradicție este demonstrată direct, integral și efectiv. Cind însă se trece la introducerea funcției propoziționale (predicat), a noțiunilor de *variabilă aparentă* și *variabilă reală* etc., problema completitudinii sistemului nu mai poate fi rezolvată atât de ușor, ceea ce înseamnă că de fapt sistemul obligă să admitem unele concepte fără definiție (primitive) și alte propoziții fără demonstrație (axiome) numai dintr-o necesitate pur logică, pe care o implică înseși procedeele de definiție și demonstrație alese și nu dintr-o necesitate reală. Or, dacă o știință admite *conventional* concepte și propoziții primitive înseamnă că ea se învirtește într-un cerc vicios ! Logica formală admite acest lucru. La rîndul lor, toate științele particulare se sprijină pe logică și totuși *ele există efectiv* — dovedă progresul matematicii, fizicii, astronomiei etc. Care este explicația ? Pentru a o oferi, logicienii sunt obligați să recurgă la „ceva” din afara logicii. Dacă o știință nu se poate dezvolta efectiv decât plecind de la primitive și axiome, rezultă că acestea din urmă aparțin științei în cauză și există o *facultate a intelectului nostru* prin care aceste elemente sunt sesizate direct, nemijlocit. Cum se produce de fapt acest fenomen ? *După părerea noastră, din sesizarea directă, senzorială a unui obiect, intelectul oprește numai o anumită reprezentare (formală) a universalului. Aceasta se dezvoltă și se completează pe parcurs, poate intra în combinație cu alte universale (esențe) și poate avea o existență și o viață proprie a cărui suport este, de astă dată, creierul uman și nu obiectul în sine. Măsura în care anumite stări, situații și procese din intelect corespund — prin verificări practice, experimentale, voite sau întîmplătoare — cu stările, situațiile și procesele reale constatăte cu ajutorul senzațiilor de-a lungul vieții, ne fac să gîndim că am obținut o imagine (teorie, descoperire etc.) mai apropiată sau mai depărtată de realitate, deci de adevăr, cu alte cuvinte o construcție științifică corectă.*

Iată-ne, aşadar, întorsî de unde am plecat, la acea misterioasă „operație a intelectului”, inteligența indivizibilelor, a esențelor, a categoriilor lui Aristotel, a universalului (*informatio intellectus*) — reconsiderată astăzi, de către unii autori, ca premisă a logicii moderne ([27] p. 363).

Dar nu numai procesul dobîndirii esențelor, a universalului, rămîne învăluit de mister ci și acela al compunerii și diviziunii formelor, a esenței, „care are loc printr-o acțiune specială a intelectului de a uni și separa”, cum spune Aristotel, de analiză și sinteză care duce la judecată, la stabilirea adevărului și falsului (*logos apophanticos*). Reducînd unirea sau separarea numai la combinarea unor semne golite de orice conținut, logica matematică a abandonat de fapt funcția creațoare a gîndirii.

Dacă logica formală nu este capabilă să modeleze prin ea însăși gîndirea, să vedem ce ne oferă psihologia și biologia.

„Inteligenta este aptitudinea de a sesiza anumite raporturi. Această sesizare poate fi sau senzorială (ca la animale) și în acest caz ea se dato-

rează reflexelor condiționate, sau *intellectuală* (ca la om), și aici intervin *limbajul și conceptele*" ([80] p. 182).

Așadar, din nou despre concepte. Numai că, de astă dată ele se integrează în comportarea generală *intellectuală* a întregului organism, în ceea ce psihologiei numesc *conduită*. Conform unor teorii întâlnite în lucrările de specialitate, conduită ar reprezenta cîmpul total al subiectului în raport cu obiectele înconjurătoare, sentimentele — dinamica acestui cîmp, iar percepția, motricitatea și inteligența ar depinde exclusiv de structura acestui cîmp ([68] p. 57—64).

Psihologia este prin definiție o știință bazată pe observație și experiment. În psihologie, un experiment valorează cît o mie de teorii. Din nefericire însă, înțelegerea mentalității umane nu se poate atinge numai prin experimente. Psihologia se sprijină pe logică și biologie. Logica furnizează aparatul matematic necesar descrierii raționamentelor aşa cum lingvistica oferă o descriere a limbajului. Care este însă conținutul deducției, al inferenței logice? În ultimii ani psihologiei au utilizat teoria informației, teoria detectării semnalelor, teoria reglării automate, holografia, gramatica transformațională și concepte preluate din programarea calculatoarelor ([53] p. 1), fără a reuși însă să găsească răspuns la această întrebare.

Numărul celulelor nervoase (neuronilor) din creierul uman este de ordinul $7 \cdot 10^9$. Să cotăm că fiecare neuron ar putea stabili pînă la 600 000 conexiuni nervoase cu celule învecinate, numărul total al conexiunilor posibile ar ajunge la $10^{2783\,000}$ ([63] p. 266). Simpla constatare a uriașei puteri combinatorice a creierului nu ne ajută însă să avansăm în desco- perirea mecanismului gîndirii, aşa cum nu ne ajută la elucidarea acestui fenomen nici perfecta cunoaștere a structurii unei celule nervoase. După opinia unor savanți de renume mondial în neurochirurgie, neuronii ar fi doar niște simpli acumulatori și emițători de energie, comparabili cu microprocesoarele din tehnica de calcul. și aşa cum o secțiune prin calculator nu oferă informații asupra programelor care se derulează în momentul respectiv în sistemul de calcul, nici o secțiune prin creier nu poate oferi mai mult. După aceiași autori, procesul care se desfășoară în creierul uman are un înalt *caracter probabilistic*, „... și nouă ni se pare că tocmai acest caracter probabilistic al legăturii dintre *input* și *output*, dintre stimuli, structura de competență și performanță, *constituie specificul și esența inteligenței*" [9]. Această formulare, cu totul remarcabilă, nu ne face să avansăm totuși prea mult în problema noastră. Alte ipoteze, deosebit de fascinante, ca aceea a emisiei de cîmpuri bioenergetice de către toate celulele, nu numai a celor specializate în gîndire, și stabilirea unei intercomunicații coordonatoare pe baza unor coduri interne (încă nedescoperite, dar constataate cu ocazia diverselor experiențe) [21] nu fac decît să incite curiozitatea și imaginația, fără a oferi, deocamdată, soluția mult așteptată.

Mult mai apropiate de posibilitățile oferite de stadiul actual al cunoașterii și, în consecință, cu perspective imediate de experimentare și dezvoltare, ni se par *încercările de rezolvare a unor probleme complexe de tip uman cu ajutorul calculatoarelor*, activitate care intră în sfera de preocupări a Inteligenței Artificiale (I.A.).

„Calculatorul este un sistem fizic, dar din punct de vedere logic nu are nici o importanță dacă el este constituit din relee, ventile, tranzistori, sau circuite integrate. Dacă circuitele sale sunt adecvate atunci el poate



imită comportamentul oricărui alt calculator (lucru realizat — N.N.) și, de ce nu, chiar al creierului uman (lucru nerealizat încă — N.N.). Ceea ce contează nu este construcția sa fizică ci *logica operațiilor* pe care le execută. Calculele cele mai complexe pot fi declanșate de o sevență corespunzătoare de semnale electrice denumită, în mod obișnuit, *program*. Creierul este și el un sistem fizic, într-un anume fel organizat și, probabil, operațiile mentale nu sunt decit niște simple „calcule” care depind nu atât de fiziologia celulelor nervase cît de logica programului pe care îl execută” ([54] p. 7 — 8). Totuși, această logică pare a se lăsa greu descoperită, deși de la primul program scris de către Newell, Shaw și Simon în 1958 pentru a verifica teoreme logice și care constituie de fapt prima încercare de simulare a gîndirii umane, pînă în prezent, au trecut mai bine de 25 ani. Fără îndoială însă că între timp s-au făcut progrese remarcabile. Testul *Turing* — aprecierea de către un observator uman dacă răspunsurile imprimate pe un teletype în urma unei întrebări oarecare au fost date de calculator sau de o ființă umană — este considerat astăzi a fi prea ușor. Dovada — programul *ELIZA*, construit de J. Weizenbaum, care în 1967 cu versiunea *DOCTOR* a reușit să inducă în eroare chiar și un expert în materie*! ([20] p. 225 — 226). Rezultate promîțătoare au fost obținute în multe domenii importante, cum ar spre exemplu: sistemele de reprezentare, limbajele și modelele pentru roboți [16, 50, 73, 74], formalismele pentru cunoaștere [76, 77] ([62] p. 355), construirea imaginilor spațiale ([51] p. 89 — 97), progresele în înțelegerea limbajului natural [22, 78] și.a. Unele scrimeri, apărute și la noi în țară în ultimul timp în acest domeniu, atestă preocupația din ce în ce mai largă a oamenilor de știință și specialiștilor pentru rezolvarea unor probleme practice complexe de producție cu participarea cît mai eficientă și la un nivel de implementare superior a tehnicii de calcul.

Abordarea multidisciplinară socio-economică și tehnică, cu un înalt profesionalism și spirit de răspundere față de nevoile colectivității în etapa istorică pe care o parcurgem, a problemelor practice privind realizarea rapidă a unor produse industriale, cu o tehnologie elastică, capabilă să facă față cerințelor variate ale pieții externe, care să înglobeze un consum minim de energie și materii prime dar totodată maximum de inteligență și prelucrare, constituie cerințe stringente ale economiei noastre socialiste de supraviețuire în contextul unei civilizații de supertehnicitate promovată astăzi pe plan mondial, grevată de puternice crize economice și politice.

Promovarea pe scară largă a microelectronicii și tehnicii de calcul [24] — cu alte cuvinte a produselor industriale care au înglobate în ele însele o înaltă tehnicitate și care sunt utilizate direct în automatizarea celor mai complexe prelucrări ale materiei — constituie premisa dezvoltării și în țara noastră a unui nivel superior de produse tehnice, așa-zise „inteligente”, care pot înlocui nu numai munca fizică a omului ci în mare măsură și pe cea intelectuală.

Unele încercări în acest domeniu au și fost efectuate în proiectarea asistată de calculator, bazată pe reprezentarea cunoașterii [42], noi concepții privind realizarea calculatoarelor cu utilizarea unor elemente de inteligență artificială [43] și.a. Primul Simpozion național de inteligență artificială și robotică, ținut în luna octombrie 1981 la București, a reunit

* Incidentul a fost relatat de Daniel Bobrow în ACM SIGART Newsletter din decembrie 1968.

o participare largă, de prestigiu, stîrnind un viu interes din partea publicului. S-au prezentat lucrări originale, unele chiar la nivel conceptual*), altele vizînd un domeniu restrîns de aplicabilitate — aşa-zisele *sisteme inteligeante de tip expert* [41]. Din această ultimă categorie face parte și prezenta lucrare, domeniul ales fiind acela al *conducerii operative a producției*.

1.2. Oportunitatea introducerii unor elemente de inteligență artificială în conducerea operativă a producției

Din cele de mai sus putem deduce următoarele :

1° Mecanismul gîndirii, deși a format obiectul unor studii încă din cele mai vechi timpuri, nu a fost nici pînă astăzi complet elucidat.

2° În gîndire, un rol esențial îl joacă *conceptele, categoriile*, care asigură legătura între modelele logice ale intelectului și realitatea înconjurătoare. Dintre concepte, unele sint transmîse pe cale genetică din generație în generație, altele se formează la individ în cursul vieții sale.

3° Imposibilitatea explicării mecanismului inferențial duce la dificultăți în aplicarea unor inferențe logice complexe, care intervin cel mai adesea în realitatea practică și unde abundă o serie de cuantificatori nedeciși (*unii, cîțiva, o bună parte, o mulțime de, puțin, mult* etc.).

4° Pentru a rezolva o problemă, creierul uman își realizează, cu mijloace care îi sint proprii, un model, o schemă de reprezentare și de acțiune. Se constată însă că nici o problemă nu se prezintă izolat. Pentru reprezentarea și rezolvarea sa, se utilizează concepte, esențe. Or, se știe că nu există concept izolat, toate formînd un sistem deductiv care pornește de la un anumit număr de noțiuni primare, cu un grad mare de generalitate, și sfîrșește din aproape în aproape la concepte utilizate numai în domenii particulare de activitate. Așadar, pentru rezolvarea unei probleme cit de cit complicate este necesară de fapt reprezentarea prin model a *întregii lumi exterioare* — lucru care nu se poate face deocamdată, cu mijloacele de care dispunem, pe cale artificială.

5° Singura cale de urmat în vederea rezolvării unor probleme complexe cu ajutorul calculatorului — pînă la descoperirea unor mecanisme inferențiale generale și construirii unui model general al lumii exterioare — este *restrîngerea ariei de aplicabilitate* și reducerea dimensiunilor modelului, cu alte cuvînte realizarea unor sisteme de tip expert. Pentru un asemenea sistem va fi necesară construirea unui mecanism inferențial simplu, mergînd chiar pînă la predicate particulare care să răspundă în mod precis, prin intermediul calculelor, la problema pusă.

Dacă restrîngem sfera de acțiune la activitatea economică a unei întreprinderi și ne limităm la problemele ridicate de către conducerea operativă a producției, observăm că pornind de la definiția, într-un anume fel, a însăși acestei activități și construind un mod de rezolvare general aplicabil pentru orice problemă de conducere operativă, ajungem la o limitare suficientă pentru a putea dezvolta în acest cadră un *model inteligent de conducere operativă a producției*.

Este ceea ce își propune să realizeze prezenta lucrare.

Pentru a ajunge însă la modelul propriu-zis, este necesară lămurirea în prealabil a unor noțiuni — concepte — pe care se bazează modelul

*¹ Lucrări prezentate de M. Drăgănescu, C. Arseni, Edm. Nicolau, I. Georgescu și alții.

și anume : *proces de decizie, decident uman, alegere a alternativelor, incertitudine, programarea deciziei, multiraționalitate* și.a. (vezi capitolul 2). O simplificare esențială, care mărește considerabil șansele de realizare practică a modelului, fără a afecta calitativ procesul de conducere operativă a producției este limitarea la relațiile de tip *cauză-efect*. Potrivit acestei categorii de relații, orice *acțiune* a modelului este rezultatul alegerii unei *alternative posibile* din mulțimea de variante inscrise în acel moment în baza de cunoștințe, pe baza unor *criterii* explicitate în prealabil și aplicând *algoritmi* specifici. *Inteligenta* modelului constă în utilizarea unor algoritmi cu grad mare de generalitate, care pot fi aplicăți în rezolvarea unor clase largi de probleme privind conducerea operativă a producției în unitățile economice. Un rol esențial în structurarea și ierarhizarea algoritmilor îl are *limbajul* (vezi capitolul 3). Calitățile sale privind selectarea și prezentarea informațiilor (capitolul 3), planificarea comportamentului (capitolul 4) precum și utilizarea sa directă în cadrul modelului intelligent de conducere operativă (capitolul 5), sint hotărîtoare pentru realizarea unor modele mai complexe în care „forța” calculatorului — exprimată prin executarea unor sarcini dificile, cum ar fi aceea a ordonanțării producției (capitolul 6) — să poată fi combinată cu reducerea progresivă a gradului de nedeterminare a problemei (vezi capitolul 7).

Creînd o „lume” de dimensiuni reduse — aceea a comenziilor și informațiilor cu caracter economic legate de conducerea operativă a producției din cadrul întreprinderilor constructoare de mașini — prezenta lucrare își propune să aplice direct unele elemente de inteligență artificială, într-un domeniu de importanță majoră pentru economia noastră națională.

CAPITOLUL

2

Modele ale procesului de decizie

Dacă ne referim la activitatea economico-productivă ea se realizează printr-o succesiune de procese de decizie. Dar decizie fără informație nu există. și nici prelucrarea complexă a informației fără calculator electronic. Așadar, vom aminti despre procesul informațional-decizional și despre o serie de concepte în legătură cu decizia, fără a avea pretenția de a epuiza acest subiect care el însuși se pretează la o abordare mult mai amplă, de sine stătătoare. Scopul pentru care menționăm aici unele teorii și în special modele matematice cu aplicații directe în practica economică a întreprinderilor este acela de a înlesni înțelegerea capitolelor următoare în care se face apel la proceduri automate construite pe baza acestor modele. Dinamica deciziei, programarea deciziei, tabele de decizie, conceptul de utilitate, metoda ELECTRE, metode de clasificare, conceptul de multirationalitate, constituie cîteva din subtitlurile cele mai reprezentative ale acestui capitol.

2.1. Considerații preliminare

Complexitatea activității umane, modul conștient de a gindi și acționa în diverse împrejurări, implică la tot pasul elemente de decizie.

Decizia nu reprezintă un simplu „moment”, ea este un proces evolutiv continuu și în spirală ascendentă (fig. 1).

Majoritatea cercetătorilor leagă teoria deciziilor de teoria informației, cel puțin prin aceea că fundamentele actului decizional îl constituie disponerea în prealabil de informații. Acest lucru ne determină să numim în continuare procesul de decizie proces *informațional-decizional*. Fondul informațional trebuie să cuprindă multimea tuturor alternativelor, iar în măsura posibilului și toate consecințele lor imediate. Desigur, într-un proces dinamic, consecințele nu pot fi totdeauna cunoscute, ceea ce limitează de la început orice teorie. Pentru a se putea trece totuși la studiul matematic al deciziei se face presupunerea că în orice situație se va opta pentru una din variantele ale cărei consecințe sunt știute și pot fi evaluate.

2.2. Formalizarea și studiul matematic al deciziei

Un proces decizional este caracterizat, în general, prin multimea variantelor (V_i), $i = 1, \dots, m$, multimea criteriilor (C_j), $j = 1, \dots, n$, multimea stărilor naturii (S_k), $k = 1, \dots, p$ și matricea consecințelor $X = (x_{ijk})$, $i = 1, \dots, m$, $j = 1, \dots, n$, $k = 1, \dots, p$.

Se consideră următoarele clase de decizie stabilite de măsura cunoștințăii factorilor determinanți pentru selectarea unei variante:

a. *Decizii în condiții de certitudine*. Să presupun cunoscute toate alternativele de acțiuni posibile, și determinate strict consecințele oricăreia din selecții.

b. *Decizii în condiții de risc*. Se presupun, de asemenea, cunoscute toate alternativele posibile de acționare, în schimb consecințele lor nu sunt unice, ci se circumscriz într-o mulțime (de obicei finită) căreia i se asociază un cimp de probabilitate.

c. *Decizii în condiții de incertitudine*. Se cunosc toate alternativele posibile de acționare, dar deși se cunosc consecințele fiecărei variante, nu este cunoscută probabilitatea de apariție a acestor consecințe.

Se consideră că un proces de decizie este caracterizat prin mai multe coordonate definitorii [11]:

a. *decidentul* definit prin elementul care efectuează operația de selecție a uneia din alternativele posibile;

b. *formularea* pe care decidentul o atribuie obiectului actului decizional;

c. *mulțimea alternativelor* posibile supusă selecției decidentului;

d. *mulțimea criteriilor de decizie* de care dispune decidentul pentru selectarea unei alternative;

e. *mulțimea stărilor naturii*, de fapt a sistemului pentru a căruia conducere se ia decizia definită ca mulțime a dependențelor alternativă-consecință;

f. *mulțimea consecințelor* fiecăreia din alternative asupra criteriilor de decizie;

g. *obiectivele decidentului* care se doresc a se atinge printr-o anumită politică a selecției alternativelor, politică exprimată prin anumite restricții asupra valorilor criteriilor de decizie, deci a consecințelor actului decizional.

Cu aceste elemente de limbaj, teoria deciziei în sine a fost elaborată și s-a dezvoltat în diverse direcții orientate fie spre luarea în considerare a teoriei utilității, fie spre teoria statistică a deciziei, teoria deciziilor multi-dimensionale (cu mai multe criterii) sau teoria deciziilor de grup.

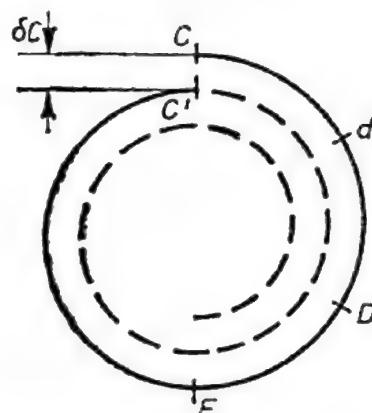
2.3. Dinamica procesului

Procesul de decizie cuprinde, în mod obligatoriu trei faze [37] :

- (A) Decizia (reprezentată, pe figura 1, de segmentul *CD*).
- (B) Execuția (reprezentată de segmentul *DE*).
- (C) Controlul (reprezentat de segmentul *EC'*).

În cadrul figurii 1, punctul *C* reprezintă punctul inițial de declanșare a acțiunii de atingere a unui obiectiv (concepția), iar punctul *E* reprezintă punctul de realizare finală (execuția).

Fig. 1. — Schema procesului de decizie : *C* — concepția ; *d* — deliberarea ; *D* — luarea deciziei ; *E* — execuția ; *C'* — controlul ; δC — abaterea obiectivului realizat față de obiectivul planificat.



Punctele *d* și *D* reprezintă evaluarea modalităților de acțiune (deliberarea), respectiv luarea deciziei (alegerea unei alternative și transmisarea ei spre executare).

Terminarea executării este urmată de verificarea atingerii obiectivului *C'* prin compararea cu obiectivul propus *C*. Aceasta este faza de control, în urma căreia se trece la o nouă acțiune, în funcție de mărimea abaterii δC .

Procesul continuă ca mai înainte urmând o trajectorie în spirală, sau se oprește în momentul în care există coincidență între *C* și *C'*.

(A) *DECIZIA*. Prima fază (A) în urma căreia este formulată decizia, se compune la rîndul ei din următoarele etape : (a) pregătirea deciziei, (b) deliberarea, (c), adoptarea soluției.

(a) Prin *pregătirea deciziei* se înțelege acțiunea de colectare, verificare, sintetizare și sistematizare a informațiilor care *contribuie*, într-un fel sau altul, la construirea soluțiilor posibile.

Pentru scurtarea timpului de pregătire a deciziei, utilizarea mijloacelor de prelucrare automată a datelor devine obligatorie.

(b) *Deliberarea* se compune din : construirea unui număr de soluții posibile ; verificarea condițiilor de aplicabilitate ale fiecărei soluții pentru perioada de timp și în cazul concret analizat ; confruntarea punctelor de vedere diferite ale persoanelor care participă la deliberare, cu luarea în considerare a aspectelor de ordin psihologic, sociologic, istoric, geografic etc. Aceste aspecte, considerate auxiliare în raport cu criteriul tehnico-economic, pot juca, în anumite situații, un rol de prim ordin.

(c) *Adoptarea* soluției, are în vedere alegerea uneia din variantele propuse și discutate în faza anterioară, variantă care întrunește majoritatea sufragiilor persoanelor participante la deliberare.

(B) *EXECUȚIA*. Cea de-a doua fază, se referă la punerea în mișcare a mijloacelor materiale și umane disponibile, conform programului de acțiune adoptat. Este o fază de transformări materiale în care conducederea directă a proceselor are rolul principal. Mijloacele de automatizare, calculatoarele de proces, pot participa direct la această fază.

(C) *CONTROLUL*. Ultima fază în procesul de decizie nu marchează totdeauna sfîrșitul procesului — care poate fi reiterat pe o treaptă superioară.

Controlul constă în colectarea informațiilor referitoare la realizarea obiectivului. Prin compararea acestuia din urmă cu obiectivul propus se stabilește mărimea și natura abaterii, se analizează cauzele care au dus la nerealizarea obiectivului, se iau măsuri de recuperare a întîrzierilor și corectare a abaterilor în viitorul ciclu : decizie — execuție — control. Dacă abaterile sunt importante, se trece la realizarea și replanificarea întregii acțiuni — ceea ce constituie de fapt un nou proces de decizie.

2.4. Decizii cu caracter de rutină

Majoritatea modelelor de decizie pleacă de la premise de caracterizare statistică a mediului decizional, premise care nu se întâlnesc ca atare în toate situațiile. De asemenea, într-o serie de cazuri nu se poate vorbi de un act decizional în sensul cel mai propriu al cuvîntului.

Fie o situație de decizie, caracterizată prin următoarele trăsături particolare :

1) Există o mulțime \mathcal{A} de alternative (a_j) , $j = 1, \dots, n$, constituite din suite de acțiuni care pot fi întreprinse pentru atingerea unor obiective prestabilite.

2) Există, de asemenea, o mulțime C de condiții (C_l) , $l = 1, \dots, p$. Fiecare condiție l poate fi (teoretic) indeplinită, în care caz starea ei s_l se notează cu 1 sau, dimpotrivă, poate rămîne neîndeplinită, în care caz starea respectivă se notează cu 0. Parametrul s_l poate lua, de asemenea și o serie de valori intermediare cuprinse între 0 și 1.

3) Se consideră vectorul parametrilor de stare $(s_l)_l$, $l = 1, \dots, p$ care caracterizează la un moment i gradul de îndeplinire al celor p condiții. Acest vector definește o stare a naturii S_i . Există, evident, atîțea stări ale naturii posibile cîte combinații de p parametri (s_l) , $l = 1, \dots, p$ distințe se pot forma.

Observație. Pentru cazul particular în care parametrul s_l nu poate lua decît două valori, 0 sau 1, numărul maxim al stărilor este 2^p .



Adoptarea unei alternative a , în starea naturii S_i , caracterizată prin vectorul de stare $(s_l)_i$, $l = 1, \dots, p$ poate avea, în general, o consecință favorabilă sau mai puțin favorabilă din punctul de vedere al obiectivului urmărit. Se definește un criteriu unic IS_i , denumit indice de selecție pentru starea i , care asigură alegerea unei singure alternative a_{j*} , cea mai favorabilă, în funcție de multimea consecințelor adoptării fiecărei alternative. Aceasta înseamnă că pentru fiecare alternativă j se calculează, în starea i , o consecință, se compară această consecință cu criteriul IS_i și se alege alternativa $j^* \in \{1, \dots, n\}$ cea mai favorabilă.

De exemplu, pentru a calcula consecința unei alternative j , se definește mai întâi vectorul de stare cu consecința cea mai favorabilă pentru acea alternativă, fie acesta $(s_l)_j$, $l = 1, \dots, p$ în care parametrii s_l au valorile 0 sau 1. Se definește apoi distanța d între un vector de stare i oarecare și vectorul de stare cu consecința cea mai favorabilă j

$$d_{ij} = \left[\sum_{l=1}^p (s_{li} - s_{lj})^2 \right]^{1/2}, \quad l = 1, \dots, p.$$

Pentru starea S_i se calculează distanțele $(d_{ij})_i$, $j = 1, \dots, n$ între vectorul ce caracterizează starea i $(s_l)_i$, $l = 1, \dots, p$ și vectorii de stare cu consecințele cele mai favorabile pentru fiecare alternativă $((s_l)_j)$, $l = 1, \dots, p$, $j = 1, \dots, n$.

Criteriul de selecție IS_i este exprimat prin formula

$$IS_i = \{j^* \mid d_{j^*i} = \min (d_{ij})_i, \quad j = 1, \dots, n\}.$$

Se ajunge astfel ca, pentru un ansamblu finit S de stări ale naturii S_i , $i = 1, \dots, m$ să putem lua decizii certe asociind fiecărei stări S_i o alternativă a , unic determinată.

Imperativul luării deciziei a , în conjunctura S_i este echivalentul exprimării condiționate.

Dacă s_{1i} și s_{2i} și ... și s_{pi} atunci execută acțiunea a_j , caracterizată printr-o suită de operații subordonate obiectivului propus.

Aceasta înseamnă că în general, selecția unei anume alternative prin evaluarea condițiilor într-o anume situație strict definibilă este în mod riguros unică.

Desigur, se poate considera că un asemenea caz se încadrează în tabloul tradițional oferit de teoria generală a deciziei, fiind un exemplu de decizie în condiții de certitudine. În acest moment accentul cade nu atât pe actul decizional ca atare, ci pe identificare, respectiv cunoașterea stării naturii S_i . Cu alte cuvinte, se impune un proces de informare, de investigare a naturii, a realității pentru cunoșterea precisă, pentru determinarea completă a stării S_i .

Efectele măsurilor respective vor fi și ele controlabile. În acest sens se poate vorbi de o programare a actului decizional prin selectarea unor alternative prestabilite de moduri de acționare.

Riguros vorbind, într-un asemenea caz actul decizional și de conducedere capătă un caracter de rutină, nereclamind nici un efort creator. Cu toate acestea nu se poate spune că este mai puțin important, deoarece el este complicat de o serie de elemente cum ar fi: existența unui număr foarte mare de alternative; existența unui număr considerabil de condiții

și evaluarea acestora ; dificultatea depistării prealabile a tuturor varianteelor de acționare și a tuturor combinațiilor posibile și semnificative de valori ale parametrilor de stare ; necesitatea adoptării unei decizii rapide, ceea ce înseamnă necesitatea unei analize rapide și exacte a situației existente și a unei evaluări a criteriilor de selecție într-un timp cît mai scurt cu puțință, precum și a depistării imediate a alternativei ce corespunde realizării din acel moment a acestor criterii.

Generalizind cele enunțate mai sus, actul decizional nu este reclamat numai de sesizarea unor abateri, dar chiar și de identificarea unor evenimente cum ar fi : inceputul/sfîrșitul unor perioade de plan ; cereri de informare din partea conducei ; inceputul/sfîrșitul unor fenomene sau procese din sistem sau din afara sistemului, dar cu implicații asupra funcționării sale ; modificarea unor obiective inițiale etc.

2.5. Programarea actului decizional

Măsurile cuprinse în cadrul diverselor alternative sunt denumite de multe ori *acțiuni* care trebuie întreprinse. Criteriile de alegere a acestor alternative se pot formula prin intermediul așa-numitelor *condiții* care rezumă toate situațiile considerate ca semnificative pentru actul decizional.

Legarea unei anumite situații, caracterizate prin realizările concrete ale factorilor de influență, de o anume variantă de adoptat va constitui o *regulă* pentru decident.

În legătură cu acest considerent merită să se aibă în vedere unele aspecte :

1. Primul se referă la specificarea legăturii intime condiții-acțiuni, deci la reguli ca atare.

2. Un al doilea are în vedere procesul insuși de selecție, ceea ce reclamă :

a) Identificarea situației prezente prin evaluarea condițiilor ce o caracterizează și pot conduce la adoptarea unei anume politici. Cu alte cuvinte, se impune o analiză a acestei situații care să conducă la examinarea numai a celor condiții care sunt semnificative în contextul dat.

b) Existența și elaborarea unei strategii de selecție care să permită recunoașterea efectivă a corespondenței deterministe condiții-program de măsuri (acțiuni) incifrate în regulă.

Funcție de importanță care într-un moment sau altul se acordă unuia sau altuia din aspectele amintite mai sus, există mai multe moduri de a reprezenta legătura intimă, prin reguli, între condiții și programele de acțiuni. Acestea sunt : metoda narativă, metoda organigramei, metoda tabulară, metoda tabelei de decizie. Dintre metodele [29] enumerate mai sus, ne vom opri în cele ce urmează, asupra conceptului tabelelor de decizie.

În literatura de specialitate există multe controverse asupra oportunității utilizării tabelelor de decizie și a gradului de extindere a acestei utilizări în diverse domenii de activitate. Unii autori le consideră ca un panaceu pentru toate problemele, alții le contestă orice utilitate. Desigur, nu este firesc să se credă că ele pot rezolva toate problemele complexe care se pun în fața unui conducător, responsabil de bunul mers al activității din compartimentul încredințat. Importantă este înțelegerea a ceea ce reprezintă în primul rînd aceste tabele și anume, un mijloc de a figura criteriile combinate de selecție ale unor alternative de acțiuni, în cazul

unui comportament strict determinat al proceselor din cadrul compartimentului studiat.

Privită din acest punct de vedere, decizia, deosebită de caracterul ei creativ poate fi asociată nu numai actului de conducere administrativă, dar chiar și unei „conduceri” a unui proces mai puțin concret în aparență. De aici și posibilitatea utilizării acestor concepte în informatică unde probleme de selecție deterministă a unor variante de măsuri apar în diversele faze ale conducerii operative a producției, efectuându-se doar echivalările necesare. Astfel în anumite situații vor fi acționate anumite proceduri componente ale unui sistem informatic pentru conducerea producției.

În ultimă instanță se pot evidenția următoarele virtuți principale ale tabelelor de decizie:

1) Prezintă un mod de comunicație între diversii specialiști informaticieni, sintetic și ușor de înțeles. *Organograma* nu oferă suficiente mijloace de standardizare, fiind greu aplicabilă pentru anumite structuri de prelucrări și datorită conceptului ei *secvențial* nu este compatibilă cu o situație combinativă.

Prin mijlocirea talelei de decizie se prezintă sub forma regulilor *o decizie logică* oricără de complexă, această prezentare putându-se face dintr-o dată indicind explicit tot ansamblul situațiilor care conduc la o anume succesiune de acțiuni și nu la alta.

2) Tabela de decizie se prezintă ca un mijloc eficace de analiză. Ea oferă posibilitatea verificării luării efective în considerare a tuturor situațiilor posibile, deoarece construcția însăși a talelei de decizie obligă la cuprinderea în ea a tuturor combinațiilor de situații elementare distincte. Astfel, deseori omisiuni ale unor cazuri, nu totdeauna nesemnificative, întâlnite în procesul de analiză, pot fi evitate.

3) Tabelele de decizie includ în ele conceptul de modularitate.

4) Tabela de decizie care descrie o procedură de prelucrare la nivel de program calculator poate fi relativ ușor convertită în programul însuși, utilizând tehnici manuale sau chiar asistate de către calculator (deci automate), prin mijlocirea precompilatoarelor.

2.6. Conceptul de utilitate. Metoda ELECTRE aplicată pe terminal

Conceptul de utilitate a fost definit axiomatic de von Neumann și Morgenstern încă din anul 1947, în felul următor.

Axioma 1. Două variante decizionale, V_i și V_j , pot fi totdeauna comparate între ele, decindându-se dacă:

- preferă pe V_i lui V_j ($V_i > V_j$);
- preferă pe V_j lui V_i ($V_j > V_i$);
- cele două variante îi sunt indiferente ($V_i \sim V_j$).

Axioma 2. Relația de preferință este tranzitivă

$$V_i > V_j \text{ și } V_j > V_k \Rightarrow V_i > V_k,$$

iar relația de indiferență este tranzitivă și simetrică :

$$V_i \sim V_j \text{ și } V_j \sim V_k \Rightarrow V_i \sim V_k,$$

$$V_i \sim V_j \Rightarrow V_j \sim V_i.$$

Axioma 3. În afara mulțimii $V = \{V_1, \dots, V_m\}$ a variantelor simple, decidentul poate lua în considerare un tip special de variante, numite *mixturi probabilistice*, de forma

$$V' = [pV_i, (1-p)V_j],$$

în care p este probabilitatea realizării variantei V_i , $1-p$ este probabilitatea realizării variantei V_j .

Axioma 4. Dacă o variantă V_i este preferată altei variante V_j , atunci o mixtură $[pV_i, (1-p)V_k]$ va fi totdeauna preferată mixturii $[pV_i, (1-p)V_k]$, oricare ar fi varianta V_k luată în considerare.

În aceste condiții se definește *funcția de utilitate* u , pe mulțimea variantelor, cu valori în mulțimea numerelor reale

$$u : \mathfrak{V} \rightarrow \mathfrak{R}, \text{ în care mulțimea}$$

$$\mathfrak{V} = \{V, [pV_i, (1-p)V_j]\}, i, j \in \{1, \dots, m\}, i \neq j$$

și are următoarele proprietăți :

i. Fie V_i, V_j , două variante decizionale

$$V_i > V_j \Leftrightarrow u(V_i) > u(V_j).$$

ii. $u[pV_i, (1-p)V_j] = p \cdot u(V_i) + (1-p) \cdot u(V_j).$

iii. Dacă u posedă proprietățile i și ii, de mai sus, atunci ea poate suferi o transformare liniară pozitivă

$$u(V_j) = a \cdot u(V_i) + b,$$

în care $a > 0$, $b > 0$ sint numere reale și în urma căreia proprietățile i și ii se păstrează.

Procedeul practic de calcul al utilităților constă în a considera cunoscute utilitățile a două variante (simple) aparținind mulțimii V . Utilitatea celorlalte variante se determină cu ajutorul proprietății ii și a probabilităților p estimate de decident.

De exemplu, fie procesul decizional multidimensional constituit din m variante V_i , $i = 1, \dots, m$ și n criterii C_j , $j = 1, \dots, n$.

În condiția în care criteriile sint independente, se aplică metoda Neumann-Morgenstern tuturor elementelor matricei X pentru fiecare criteriu în parte. Considerind cunoscute două dintre utilități, să presupunem $u(x_{1j}) = a$ și $u(x_{mj}) = b$, relativ la criteriul j , celelalte se află prin mixturi probabilistice de tipul

$$u(x_{ij}) = p_{ij} \cdot u(x_{1j}) + (1-p_{ij}) \cdot u(x_{mj}),$$



în care p_{ij} — probabilitatea de adoptare a variantei i din punctul de vedere al criteriului j se calculează raportind diferența, în valoare absolută, între consecința x_{ij} , și consecința optimă x_{mj} la ecartul maxim posibil $|x_{1j} - x_{mj}|$.

Utilitatea unei variante V_i va fi, în acest caz, calculată cu relația

$$u(V_i) = \sum_{j=1}^n u(x_{ij}).$$

O metodă oarecum diferită de cea bazată pe conceptul de utilitate este aceea pusă la punct de societatea franceză SEMA în 1965 și cunoscută sub denumirea de „metoda ELECTRE”.

Această metodă se bazează pe aplicarea teoriei grafurilor mulțimii variantelor (V_i) , $i = 1, \dots, m$.

Pornind de la ipoteza că, indiferent de procedeul utilizat, oricare două variante pot fi comparate între ele din punctul de vedere al unui criteriu, se stabilește pentru fiecare C_j , aplicația

$$r_j : C_j \rightarrow (a_{1j}, \dots, a_{mj}), j = 1, \dots, n,$$

în care mulțimea (a_{ij}) , $i = 1, \dots, m$ este total ordonată față de o relație de ordine valabilă pentru fiecare j^* .

Pe de altă parte, se introduc niște coeficienți globali de apreciere k , ai criteriilor C_j și se definesc doi indicatori:

— indicatorul de concordanță $c(V_i, V_k)$, care arată în ce măsură varianta V_i depășește o altă variantă V_k , și

— indicatorul de discordanță $d(V_i, V_k)$, care arată în ce măsură o variantă V_i este depășită de o altă variantă V_k , metodele de calcul fiind diferite pentru cei doi indicatori ($0 < c < 1$ și $0 < d < 1$).

Față de două „valori-prag” alese de decident, p și q ($0 < p \leq 1$ și $0 < q \leq 1$), se consideră că o variantă V_i este mai bună decât o altă variantă V_k dacă și numai dacă $c(V_i, V_k) \geq p$ și $d(V_i, V_k) \leq q$.

Perechea de valori (p, q) aleasă de decident definește o clasă pe mulțimea variantelor, clasa poate fi reprezentată de graful

$$G_{p,q} = (M; U_{p,q})$$

Două variante V_i , V_k fac parte din clasa (p, q) , cu alte cuvinte există un arc de la V_i la V_k în graf $G_{p,q}$, dacă și numai dacă $c(V_i, V_k) \geq p$ și $d(V_i, V_k) \leq q$.

Pentru $p' \geq p$ și $q' \leq q'$ graful $G_{p', q'}$, este un subgraf parțial al lui $G_{p,q}$.

Rezultă că mășorind pe p și mășorind pe q se ajunge, în cele din urmă, la un graf complet.

Pentru ca o variantă V^* să ajungă să fie mai bună decât toate celelalte, trebuie să avem relația

$$c(V^*, V_i) \geq p \text{ și } d(V^*, V_i) \leq q, \text{ pentru } i = 1, \dots, m,$$

lucru care se întimplă la un nivel suficient de scăzut, respectiv ridicat, ale valorilor limită p și respectiv q .

Prima dintre variante care ajunge să surclaseze pe toate celelalte este cea preferată.

* a_{ij} pot fi numere, calificative, situații, stări de lucruri etc.

Există programe și pachete de programe [36] construite pe baza metodelor de mai sus, care oferă decidentului, cu un efort minim, indicații asupra alegerii uneia sau alteia dintre variantele analizate. Pachetul de programe DEMCRIT, de exemplu, elaborat în cadrul catedrei de informatică a Academiei „Ștefan Gheorghiu”, utilizează un sistem de telemisie care permite rezolvarea unei probleme de decizie multicriterială de la consola unui terminal de tip DAF sau CENTRONICS legat la calculatorul FELIX-0 256 (512–1024). Programele transpun pe calculator o variantă a metodei ELECTRE clasice, variantă elaborată de Gh. Boldur. Dat fiind faptul că ne vom referi în continuare, în cadrul capitolului 4, la această metodă, ne permitem să menționăm aici pe scurt diferențele față de metoda clasice expusă mai sus. Spre deosebire de metoda clasică, în varianta îmbunătățită se ține seama de următoarele:

1° Valorile de apreciere ale variantelor, valabile pentru un criteriu oarecare j , se aleg în intervalul $[0,1]$, marginile intervalului, 0 și 1, trebuind să figureze obligatoriu în sirul de valori (a_{ij}) , $i = 1, \dots, m$.

2° Coeficienții globali de apreciere a criteriilor k , se aleg tot în intervalul $[0, 1]$, astfel încât $\sum_{j=1}^n k_j = 1$.

3° Indicele de concordanță c^* se calculează cu formula

$$c^*(V_g, V_h) = \sum_{j \in J} k_j (a_{gj} - a_{hj}),$$

în care mulțimea indicilor J este definită prin:

$$J = \{j \mid a_{gj} > a_{hj}\},$$

iar indicele de discordanță d^* se calculează cu formula

$$d^*(V_g, V_h) = \sum_{j \in J'} k_j (a_{hj} - a_{gj}),$$

în care mulțimea indicilor J' este definită prin

$$J' = \{j \mid a_{gj} \leq a_{hj}\},$$

astfel încât este valabilă relația

$$c^*(V_g, V_h) = d^*(V_h, V_g).$$

Toate celelalte considerații menționate în cadrul metodei clasice rămân valabile și în varianta îmbunătățită.

O altă metodă, formulată mai general ca soluție la o problemă de clasificare a unor m obiecte care răspund sau nu unui număr de n proprietăți [17] este următoarea:

Fie (e_{ij}) , $i = 1, \dots, m$, $j = 1, \dots, n$ matricea booleană corespunzătoare *satisfacerii*, în care

$$e_{ij} = \begin{cases} 1 & \text{dacă obiectul } O_i \text{ satisfac proprietății } P_j, \\ 0 & \text{dacă nu este indeplinită condiția de mai sus.} \end{cases}$$

Se definește o distanță pe mulțimea obiectelor (O_i) , $i = 1, \dots, m$,

$$d_{ij} = \sum_{k=1}^n \delta_k,$$

unde :

$$\delta_k = \begin{cases} 1 & \text{dacă } e_{ik} = 0 \text{ și } e_{jk} = 1, \\ & \text{sau } e_{ik} = 1 \text{ și } e_{jk} = 0, \\ 0 & \text{dacă } e_{ik} = 0 \text{ și } e_{jk} = 0, \\ & \text{sau } e_{ik} = 1 \text{ și } e_{jk} = 1. \end{cases}$$

Utilizind matricea distanțelor astfel calculate (d_{ij}) , $i, j \in \{1, \dots, m\}$ și algoritmi de clasificare pe calculatorul electronic, se ajunge la separarea unor *clase de obiecte* p , obiectele care fac parte din clasa p aflindu-se între ele la o distanță mai mică sau cel mult egală cu p , $0 < p \leq m$.

Se poate construi o reprezentare grafică, ce pune în evidență clasele de obiecte și structura lor arborescentă față de nivelele de distanță.

Slăbiciunea tuturor acestor metode constă în subiectivitatea fie a aprecierii probabilităților p și a respectării condiției de „independență” a criteriilor în cazul teoriei utilității, fie a alegerii coeficienților k , de apreciere globală a criteriilor în cazul metodei ELECTRE, fie a aprecierii gradului de importanță al distanțelor în metoda clasificărilor. Există și mijloace de a reduce gradul de subiectivitate al acestor metode, dintre care, menționăm ca deosebit de eficace dar destul de dificil de aplicat în practică, *studiul statistic* al cazurilor reale deja rezolvate pe anumite domenii restrinse și compararea soluțiilor verificate practic cu soluția teoretică obținută cu fiecare din aceste metode. Este un sistem care aplicat corect duce la eliminarea treptată a gradului de incertitudine și obținerea de soluții din ce în ce mai apropiate față de realitate.

2.7. Conceptul de multiraționalitate

Teoria referitoare la procesul de decizie s-a dezvoltat și îmbogățit continuu, apărind astfel concepte noi a căror aplicare în practică poate fi justificată sau nu. Pornind de la analiza sistemelor umane H. A. Simon, E. D. Robert, R. M. Cyert, J. G. March, J. Forrester, Feldman, Kanters, L. Sfez și alții au adus contribuții deosebite la această problemă.

În cele ce urmează ne vom referi asupra conceptului de raționalitate a deciziei, propus de March și Simon în lucrarea lor referitoare la organizații [60]. Cei doi adepti ai școlii moderne de organizare consideră că decizia simplă, deterministă și categorică în alegerea unor alternative nu exprimă realist cazurile concrete întâlnite mai cu seamă în practica de conducere a întreprinderilor și propun consacrarea unor termeni ca: *alegere satisfăcătoare, scop secundar, grad de absorbie a incertitudinii, inovare, raționalitate limitată*.

Teoria lui March și Simon, precum și cele ale lui Forrester [34], Feldman și Kanters [33], Cyert și March [24] și alții, largesc de fapt conceptul matematic de decizie, așa cum este el prezentat de teoria utilităților, metoda ELECTRE și altele de același gen.

În viziunea lui Sfez, însă, teoriile menționate mai sus suferă de un mare neajuns: ele nu admit o schimbare a scopului propus inițial, o critică a finalității [70]. Noțiunea de decizie pe care o propune Sfez, în schimb, pornește de la ideea că *fiecare subsistem are propria sa raționalitate*.

Privită la nivelul întregului sistem, decizia apare astfel ca *multiratională* sau altfel spus ca o decizie reală, aplicată la o multitudine de situații concrete, specifice oricărui subsistem.

Tinând seama de premise, conceptul de multirationalitate este pe deplin justificat. Aplicarea sa în practică nefiind însă suficient de elaborată pînă în prezent [39], el nu prezintă decît o valoare pur teoretică.

Amintim aceste teorii pentru a nu da impresia că abordările matematice prezentate în cadrul acestui capitol epiuzează problema deciziei. Rationalitatea criteriilor, în general, nu trebuie privită ca pe o dogmă. Ceea ce astăzi pare a fi rațional, mîine s-ar putea să se dovedească o aberație. Concluzia este că atunci cînd se fixează asemenea raționalități calculatorului, este neapărat necesar ca ele să fie *condiționate și revizuite periodic*. Ne gîndim în special la procedurile generale de acțiune care formează cadrul de rezolvare a unor clase întregi de probleme, cum ar fi, de exemplu, aceea a conducerii operative a producției.



CAPITOLUL

3

Modelul relațional «INTELEC» de reprezentare și manipulare a cunoștințelor. Definiție, structură, realizare informatică și mod de utilizare

Noțiunile, cu caracter general sau particular, se exprimă prin siruri de cuvinte. Deși în practica vorbirii curente cuvintele și expresiile pot avea mai multe înțelesuri, pentru a conduce activitatea economică a întreprinderilor avem nevoie de un limbaj mai restrictiv, în care entitățile și frazele să fie unic determinate. Un astfel de limbaj — INTELEC — destinat conducerii operative a producției este definit în cadrul acestui capitol. Limbajul este capabil să descrie exact situațiile economice, să efectueze calcule, să dea comenzi, să execute programe și să comunice cu utilizatorul prin intermediul terminalelor de teletransmisie cerînd și completînd informațiile necesare. Suportul informatic al limbajului este o bază de date relațională (BDR) dotată cu o structură flexibilă care permite adăugarea „din mers” a unor noi caracte-ristici fără a afecta cu nimic informațiile existente. Modul de utilizare al BDR prin intermediul limbajului INTELEC este ilustrat printr-o serie de exemple. Se arată procedura de introducere a informațiilor de detaliu, plecînd de la o frază cu grad mai ridicat de generalitate considerată drept prototip generator. Se arată procedura directă de regăsire — pe baza combinării cuvintelor „cheie” — ca și o serie de pro-ceduri indirekte: prin utilizarea predicatelor [ESTE (CAZ PARTICULAR AL)], [ESTE (IN ACEST CAZ)], [SE COMPUNE DIN (ANSAMBLU)], [SE COMPUNE DIN (SUCCESIUNE)] și a. pe baza prelucrării cadrelor relaționale, stabilirea unor legă-turi logice între două informații oarecare din BDR, generalizarea unor cazuri parti-culare și crearea unor noi informații (sintetice).

3.1. Generalități

1. Numim *entitate* și o notăm cu e , o noțiune cu caracter general sau particular exprimată printr-o succesiune de cuvinte (la limită constituită dintr-un singur cuvînt), avînd un înțeles bine determinat. Entitatea are o *valoare semantică unică*, cu alte cuvinte, există o aplicație σ , injectivă, din mulțimea entităților \mathbb{S} către mulțimea semnificațiilor \mathbb{S}

$$\sigma : \mathbb{S} \rightarrow \mathbb{S}, \quad (3.1)$$

în așa fel încit unei entități $e \in \mathbb{S}$ să-i corespundă o semnificație unică a sa $\tilde{e} \in \mathbb{S}$ (egalitatea formală a două entități $e_1, e_2 \in \mathbb{S}$ / $e_1 = e_2$, implică neapărat coincidența semnificațiilor respective în \mathbb{S} , adică $\tilde{e}_1 = \tilde{e}_2$).

Observație. Dacă o entitate e_1 poate fi interpretată în mai mult de un singur fel (are mai multe semnificații posibile $\tilde{e}_{11}, \tilde{e}_{12}, \dots, \tilde{e}_{1n} \in \mathbb{S}$) trebuie să se adauge un număr de cuvinte care să eliminate ambiguitatea interpretării, prin acest procedeu din mulțimea (\tilde{e}_{1i}) , $i = 1, \dots, n$ răminind o singură semnificație \tilde{e}_{1p} , imagine a lui e_1 în \mathbb{S} , adică

$$\sigma(e_1) = \tilde{e}_1^* \quad (3.2)$$

2. Se definește \mathcal{R}^2 o *relație de succesiune binară* pe \mathbb{S} , ca fiind aplicația produsului cartezian $(\mathbb{S} \times \mathbb{S})$ către mulțimea \mathbb{S}

$$\mathcal{R}^2 : (\mathbb{S} \times \mathbb{S}) \rightarrow \mathbb{S} \quad (3.3)$$

în așa fel încit perechii ordonate (e_1, e_2) să-i corespundă entitatea $e_{12} \in \mathbb{S}$ formată din concatenarea ^{**} lui e_1 și e_2 adică

$$e_{12} = e_1 e_2. \quad (3.4)$$

Analog, se definește \mathcal{R}^3 ca *relație de succesiune ternară*

$$\mathcal{R}^3 : (\mathbb{S} \times \mathbb{S} \times \mathbb{S}) \rightarrow \mathbb{S} \quad (3.5)$$

în care tripletului ordonat (e_1, e_2, e_3) îi corespunde entitatea

$$e_{123} = e_1 e_2 e_3 \quad (3.6)$$

formată din concatenarea celor trei entități și, în general, *relația de succesiune n-ară*

$$\mathcal{R}^n : \underbrace{(\mathbb{S} \times \mathbb{S} \times \dots \times \mathbb{S})}_{\text{de } n \text{ ori}} \rightarrow \mathbb{S}, \quad (3.7)$$

^{*} Am renunțat la indicele p deoarece în acest caz el nu mai are rost.

^{**} Numim *concatenare*, scrierea una după alta în ordine, a celor două entități.

în care n -pletului ordonat (e_1, e_2, \dots, e_n) ii corespunde entitatea

$$e_{12\dots n} = e_1 e_2 \dots e_n. \quad (3.8)$$

3. Succesiunea $e_1 e_2 \dots e_n$ avind un înțeles bine determinat, poartă numele de *frază* și se notează cu f . Ansamblul tuturor frazelor va fi notat cu \mathcal{F} și va conține *mulțimea tuturor relațiilor* de tip $\mathcal{R}^2, \mathcal{R}^3, \dots, \mathcal{R}^n$ definite, la momentul respectiv, pe \mathcal{S} .

4. Numărul entităților componente ale unei fraze f poartă numele de *lungimea frazei* și se notează cu *lung* (f). Dacă

$$f = e_1 e_2 \dots e_n, \text{ cu alte cuvinte fraza } f \text{ este o relație} \quad (3.9)$$

n-ară, $f \in \mathcal{R}^n$, atunci, prin definiție

$$\text{lung } (f) = n. \quad (3.10)$$

5. Două fraze $f_1, f_2 \in \mathcal{F}$ exprimate prin relațiile:

$$f_1 = e_{1_1} e_{1_2} \dots e_{1_m} \in \mathcal{R}^m, \quad (3.11)$$

$$f_2 = e_{2_1} e_{2_2} \dots e_{2_n} \in \mathcal{R}^n, \quad (3.12)$$

sunt egale, dacă:

$$m = n \quad (3.13)$$

$$e_{1_1} = e_{2_1} \quad (3.14)$$

$$e_{1_2} = e_{2_2}$$

6. Două fraze $f_1, f_2 \in \mathcal{F}$ având expresiile (3.11) și, respectiv (3.12) se pot *compune*, rezultatul fiind o nouă frază $f_{12} \in \mathcal{F}$ numai dacă *entitățile celor două fraze se află în relație de succesiune*, cu alte cuvinte dacă expresia

$$e_{1_1} e_{1_2} \dots e_{1_m} e_{2_1} e_{2_2} \dots e_{2_n} \text{ pe care o notăm cu } e_{12}. \quad (3.15)$$

are o semnificație unică $\tilde{e}_{12} \in \mathcal{S}$.

În acest caz, este valabilă expresia

$$f_1 f_2 = f_{12}. \quad (3.16)$$

7. Lungimea unei fraze $f_{12} \in \mathcal{F}$ rezultată din compunerea a două fraze f_1 și f_2 având lungimile m , respectiv n , este egală cu $m + n$, cu alte cuvinte, este valabilă expresia:

$$\text{lung } (f_{12}) = \text{lung } (f_1) + \text{lung } (f_2). \quad (3.17)$$

8. Să construim acum aplicația pe care să o denumim *text*, în așa fel încât unui *grup de fraze* $f_1, f_2, \dots, f_m \in \mathcal{F}$ să-i putem asocia o entitate

$e_t \in \mathfrak{S}$ și, prin intermediul aplicației σ (vezi (1)), o semnificație unică $\tilde{e}_t \in \mathfrak{S}$. Spunem în acest caz că frazele f_1, f_2, \dots, f_m se află în *relație de tip text m-ară* și exprimăm acest lucru prin

$$\mathfrak{R}_{\mathfrak{F}}^m : \underbrace{(\mathfrak{F} \times \mathfrak{F} \times \dots \times \mathfrak{F})}_{\text{de } m \text{ ori}} \rightarrow \mathfrak{S} \quad (3.18)$$

și

$$\mathfrak{R}_{\mathfrak{F}}^m (f_1, \dots, f_m) = e_t. \quad (3.19)$$

Observații. 1° $\mathfrak{R}_{\mathfrak{F}}^m$ este o relație de ordinul întii referitor la fraze dar de ordinul al doilea relativ la entitățile componente, în sensul că fiecare frază este la rîndul ei o relație n -ară, așa cum am arătat.

2° Definițiile de la 1–7, de mai sus, pentru fraze, rămîn valabile și pentru texte, în care scop se va înlocui cuvîntul „entitate” prin „frază” și, respectiv, „frază” prin „text”. De exemplu, paragraful 4 se transformă în 4', astfel :

4'. Numărul frazelor componente ale unui text t poartă numele de *lungimea* textului și se notează cu lung (t). Dacă

$$t = f_1 f_2 \dots f_m, \quad (3.9')$$

atunci prin definiție

$$\text{lung}(t) = m. \quad (3.10')$$

9. Oricărei entități $e \in \mathfrak{S}$ i se poate asocia o serie de *caracterizatori* c_{1e}, c_{2e}, \dots etc. Să notăm mulțimea lor cu C_e .

Pe mulțimea $\mathfrak{C} = \{C_e\}, e \in \mathfrak{S}$ se pot defini mai multe *clase de caracterizatori*, fie mulțimea lor $\mathfrak{K} = \{i, v, \tau, \dots\}$. Un caracterizator aparține unei singure clase; pot exista, însă, simultan, mai mulți caracterizatori pentru o singură entitate.

Avem, deci, aplicația „clasă” prin care asociem unui caracterizator $c \in \mathfrak{C}$ o clasă $\chi \in \mathfrak{K}$ și numai una

$$\text{cls} : \mathfrak{C} \rightarrow \mathfrak{K}, \text{ astfel încit } \text{cls} (c) = \chi, c \in \mathfrak{C}, \chi \in \mathfrak{K}. \quad (3.20)$$

Aplicația „cls” partiziionează mulțimea \mathfrak{C} într-o serie de submulțimi $\mathfrak{C}_\chi \subset \mathfrak{C}$ disjuncte, cu alte cuvînte avem relațiiile :

$$\mathfrak{C} = \bigcup_{\chi \in \mathfrak{K}} \mathfrak{C}_\chi \text{ și } \bigcap_{\chi \in \mathfrak{K}} \mathfrak{C}_\chi = \emptyset. \quad (3.21)$$

Convenim să notăm un caracterizator i al entității e și care aparține submulțimii \mathfrak{C}_χ , $\chi \in \mathfrak{K}$ cu c_{ie}^χ . Avem, prin urmare, mulțimea

$$C_e = \{c_{1e}^i, c_{2e}^i, \dots, c_{1e}^v, c_{2e}^v, \dots, c_{1e}^\tau, c_{2e}^\tau, \dots\}. \quad (3.22)$$

10. Fie o submulțime finită $E \subset \mathcal{S}$ de entități e considerate ca simboluri având semnificații precise, alese de noi, și circumscrise unui domeniu de activitate (de exemplu, conduceerii operative a producției), submulțime pe care o numim în continuare alfabet.

Fie, de asemenea, $F' \subset \mathcal{S}$ mulțimea tuturor frazelor ce se pot forma cu aceste simboluri în modul arătat la paragraful 3 de mai sus, mulțime în care se include și f_0 — fraza de lungime zero.

Un limbaj este o submulțime $F \subset F'$, construită cu alfabetul E pe baza unor reguli P pe care le vom numi producții.

În cele ce urmează, o frază $f \in F$ se va considera constituită dintr-un sir finit de simboluri $e_1 e_2 \dots e_n$, dintre care unele din ele vor fi considerate terminale și aparținând mulțimii T , iar celelalte neterminale și aparținând mulțimii N . Printre simbolurile neterminale se va număra și simbolul de start S .

11. O gramatică [8] (de tip 0) este o construcție $G = (N, T, P, S)$, în care N și T sunt mulțimile finite de neterminale, respectiv de terminale, S este simbolul de start sau de frază iar P este mulțimea de producții. O producție este de forma: $\alpha \rightarrow \beta$, în care

$$\beta \in (N \cup T)^* \text{ și } \alpha \in (N \cup T)^* N (N \cup T)^*^{**}.$$

12. Se definește \Rightarrow_G o relație pe mulțimea $(N \cup T)^*$ astfel încit $\gamma_1 \Rightarrow_G \gamma_2$ dacă și numai dacă putem scrie:

$$\gamma_1 = \delta_1 \alpha \delta_2, \quad \gamma_2 = \delta_1 \beta \delta_2, \quad \alpha \rightarrow \beta \text{ fiind o producție în } P, \quad \delta_1, \delta_2 \in (N \cup T)^*$$

Cu alte cuvinte, γ_1 este în relație cu γ_2 dacă putem găsi un subșir al lui γ_1 care să reprezinte partea stingă a unei producții în P .

În general, o relație \Rightarrow_G este reflexivă, adică posedă proprietatea

$$\gamma \Rightarrow_G \gamma, \quad \forall \gamma \in (N \cup T)^* \quad (3.23)$$

și tranzitivă, adică satisfacă propoziția

$$\gamma_1 \Rightarrow_G \gamma_2 \text{ și } \gamma_2 \Rightarrow_G \gamma_3 \text{ implică obligatoriu } \gamma_1 \Rightarrow_G \gamma_3, \quad \forall \gamma_1, \gamma_2, \gamma_3 \in (N \cup T)^* \quad (3.24)$$

ășă încit se poate defini o închidere tranzitivă și reflexivă $\gamma_0 \xrightarrow{G} \gamma_n$ a relației \Rightarrow_G de la γ_0 la γ_n ca fiind o secvență de substituiri succesive de subșiruri în conformitate cu producții din P care conduce de la γ_0 la γ_n . Secvența de siruri obținută $\gamma_0, \gamma_1, \dots, \gamma_n$ astfel încit $\gamma_{i-1} \Rightarrow_G \gamma_i$, $i = 1, \dots, n$ se numește derivarea lui γ_n din γ_0 în G .

Limbajul $L(G)$ generat de G reprezintă totalitatea sirurilor obținute prin derivare, astfel încit acestea să fie în cele din urmă constituite numai din simboluri terminale.

^{**} Am notat cu $(X)^*$ mulțimea sirurilor ce se pot forma cu elementele mulțimii X .

13. Prin definiție, o gramatică G se numește independentă de context^{*)} dacă producțiile din P sint de forma

$$e_n \rightarrow \beta, \text{ unde } e_n \in N \text{ și } \beta \in (N \cup T)^*.$$
 (3.25)

În general, limbajul generat de o gramatică independentă de context se numește *limbaj independent de context*.

3.2. Definiția limbajului INTELEC de reprezentare și manipulare a cunoștințelor

În cele ce urmează vom defini un limbaj $L(G)$ independent de context.

Ne referim la o categorie restrinsă de entități $e \in E$, $E \subset \mathcal{S}$, din domeniul conducerii operative a producției și având următoarele tipuri de caracterizatori :

a. INDEX (notat cu ι). Caracterizatorii de acest tip sint și ei la rîndul lor, entități $e^i \in E$ care exprimă esențe, generalități, domenii, criterii de clasificare etc. și pe care ii numim, în mod obișnuit, *cuvinte cheie*^{**)} . Unei entități $e \in E$ i se va putea atașa setul $(e_1^i, e_2^i, \dots) \in E$ care caracterizează din anumite puncte de vedere entitatea considerată.

b. TIP (notat cu τ). Caracterizatorii de acest tip se referă la rolul entității în cadrul frazei (rolul sintactic). Aceștia sint următorii : N este noțiune (substantiv sau orice expresie care poate juca rolul de subiect sau complement), P – predicat (predicat sau orice expresie care exprimă o acțiune, stare sau existență), C – calificator (atribut sau orice expresie care determină sau întregește înțelesul unei noțiuni), O – operator (cuvint sau expresie care indică o operație aritmetică sau logică, legind două noțiuni având aceeași funcție sintactică).

În afara caracterizatorilor generali N , P , C , O , mai există încă doi care se referă la operații interne în cadrul limbajului și care sint : K este noțiune internă și Q – predicat intern, pe care ii vom defini cu ocazia construirii suportului informatic al limbajului (baza de date relațională).

Prin urmare, unei entități $e \in E$ i se va putea atașa una din caracteristicile N , P , C , O , K , Q , de tip τ .

c. NUMĂR (notat cu ν). Caracterizatorii de acest tip se referă la entitățile numerice și sint egale cu : I (dacă entitatea este un număr întreg ; valoarea numărului este înregistrată la caracteristica n^{VAL}), R (dacă entitatea este un număr zecimal ; în acest caz numărul se compune din următoarele două valori : valoarea întreagă – înregistrată la caracteristica n^{VAL} ; număr de zecimale – înregistrată la caracteristica n^{NZ}), E (dacă entitatea este un număr întreg urmat de 10 la o putere oarecare (pozitivă sau negativă) ; în acest caz numărul se compune din următoarele două valori : valoare întreagă – înregistrată la caracteristica n^{VAL} ; valoarea exponentului – înregistrată la caracteristica n^{EXP}).

d. n^{VAL} , n^{NZ} și n^{EXP} sint caracteristici numerice întregi având semnificații de la punctul c.

*) Sinonim cu gramatică de tipul 2, sau, încă, de tip BNF (Backus normal form, Backus-Naur form).

**) Termenul este larg răspîndit așa încît îl vom folosi și noi. Trebuie să remarcăm însă că fiind vorba de entități acestea pot fi formate din mai multe cuvinte din limbajul natural.

Așadar, mulțimea caracteristicilor admise pentru entitățile $e \in E$ aparținând limbajului $L(G)$ este $C \subset \mathcal{C}$ definită prin

$$C = \{e^t, N, P, C, O, K, Q, I, R, E, n^{\text{VAL}}, n^{\text{NZ}}, n^{\text{EXP}}\},$$

unei entități $e \in E$ având posibilitatea să-i atașăm:

- prin aplicația ι , un set de „cuvinte cheie”.

$$\iota : E \rightarrow E^t, \quad (3.26)$$

E^t fiind restricția mulțimii E la entitățile „cuvinte cheie”

$$e \rightsquigarrow (e_1^t, e_2^t, \dots, e_m^t);$$

- prin aplicația τ , un rol sintactic în cadrul frazei

$$\tau : E \rightarrow \{N, P, C, O, K, Q\}; \quad (3.27)$$

- prin aplicația ν un tip de număr

$$\nu : E \rightarrow \{I, R, E\} \quad (3.28)$$

și, în funcție de tipul numărului, una sau două valori din tripletul n^{VAL} , n^{NZ} , n^{EXP}

$$I \rightsquigarrow (n^{\text{VAL}}), \quad R \rightsquigarrow (n^{\text{VAL}}, n^{\text{NZ}}), \quad E \rightsquigarrow (n^{\text{VAL}}, n^{\text{EXP}}).$$

Așa cum oricărei entități i se pot atașa o serie de caracterizatori referitori la entitate, unei fraze $f \in F$ aparținând limbajului $L(G)$ i se pot asocia caracterizatori cu privire la fraza in sine ^{*)}, de exemplu: CF este caracterizator ce definește domeniul de valabilitate (gradul de generalitate) al aserțiunii exprimate prin f . Acesta poate fi: D — definiție; fraza pe care o caracterizează are valoare de adevăr și generalitate maximă, P — principiu; fraza respectivă exprimă un adevăr valabil pentru sistemul definit (orice modificare presupune schimbarea sistemului), G — generalizare; fraza exprimă o generalizare a unor cazuri particulare avute în vedere (ceea ce exprimă poate fi oricând infirmat de un caz particular neavut în vedere în momentul generalizării), A — aserțiune; fraza exprimă o afirmație despre un anumit lucru, putând fi creditată cu o probabilitate mai mare sau mai mică de adevăr.

n^{PF} — caracterizator numeric, cuprins între 0 și 100, servind la măsurarea probabilității de adevăr cu care poate fi creditată o aserțiune A (aprecierea este, evident, subiectivă).

Există, prin urmare și o mulțime a caracterizatorilor frazelor aparținând limbajului $L(G)$ $C_f \subset \mathcal{C}$, care, în exemplul de mai sus este

$$C_f = \{D, P, G, A, n^{\text{PF}}\}, \quad 0 \leq n^{\text{PF}} \leq 100, \quad [n^{\text{PF}} \in N,$$

unei fraze $f \in F$ având posibilitatea să-i atașăm:

^{*)} Vezi și considerațiile de la subcapitolul 3.1., paragraful 2.

— prin aplicația δ , un domeniu de valabilitate

$$\delta : F \rightarrow \{D, P, G, A\}, \quad (3.29)$$

— prin aplicația α , o valoare pragmatică de adevăr

$$\alpha: F \rightarrow \{n^{\text{PF}}\}$$

$$f \mapsto n, \quad n \in \{0, 1, 2, \dots, 100\}. \quad (3.30)$$

Să definim acum *gramatica* $G = (N, T, P, S)$, independentă de context, a limbajului $L(G)$.

În acest scop vom lua în considerare nu mai rolul entității în cadrul frazei (caracteristica de tip τ) precum și succesiunea entităților în cadrul frazei*. Pentru usurința scrierii vom utiliza următoarele notări:

- 1) Entitățile de tip *noțiune N*-simple sau compuse din mai multe entități — se vor încadra între semnele „<” și „>”.
- 2) Entitatea de tip *predicat P* se va încadra între semnele „[” și „]”.
- 3) Entitățile de tip *calificator C* și *operator O* se vor încadra între semnele „(” și „)”.
- 4) *Noțiunea internă K* și *predicatul intern Q* vor fi notate cu un asterisc (*) în afara ultimului separator (din dreapta) sus.

4) Notiunea internă R și predicatul intern Q vor fi notate cu un asterisc (*) în afara ultimului separator (din dreapta) sus.

Exemplu. \langle Notiune \rangle^* reprezintă notația pentru o notiune internă; $[$ Predicat $]^*$ reprezintă notația pentru un predicat intern.

5) Simbolul neterminat $\langle Text \rangle$ primește obligatoriu un index alcătuit dintr-o entitate $\langle N \rangle$ care reprezintă *titlul* textului respectiv. *Exemplu.* $\langle Text \rangle_{\langle N \rangle}$ are semnificația: $\langle Text \rangle$ avind titlul $\langle N \rangle$.

Mulțimea simbolurilor neterminale N ale limbajului $L(G)$ este următoarea: $N \equiv \{\langle Text \rangle_{\langle N \rangle}; \langle Grup\ de\ fraze \rangle; \langle Frază \rangle; \langle Grup\ ON \rangle; \langle Grup\ CN \rangle; \langle Notiune \rangle; \langle Calificator \rangle; \langle Predicat \rangle; \langle Comandă \rangle; \langle Comandă\ conditionată \rangle\}$. Simbolul de start S este $\langle Text \rangle_{\langle N \rangle}$.

Mulțimea simbolurilor terminale T este constituită din ansamblul entităților aparținând domeniului conducerii operative a producției, la care se adaugă o serie de noțiuni cu caracter general, frecvent întâlnite în limbajul natural. Toate noțiunile trebuie să fie precis definite și să aibă un înțeles unic. Parte din simbolurile terminale utilizate în cadrul exemplelor care figurează în lucrarea de față se găsesc explicate în cadrul tabelelor 1 și 2 din acest capitol.

Mulțimea predicatelor P este definită prin următoarele reguli de alcătuire a frazelor și textelor:

- I. *NPN*
- II. *NPN(ON)* sau *N(ON)PN*
- III. *CNPCN* sau *(C)NP(C)N* sau
(C)NP(C)N(O(C)N) sau *(C)N(O(C)N)P(C)N*

care reprezintă cazurile I și II în care N a fost înlocuit cu grupul (CN) iar C poate fi repetitiv (C)

* Toate noțiunile situate înaintea predicatului sunt considerate subiect iar cele situate după predicat sunt considerate complement.

IV. $\langle SE \rangle \cdot P(C)N$ sau $\langle SE \rangle \cdot P(C)N(O(C)N)$

care reprezintă cazul particular în care subiectul este fix și notat cu $\langle SE \rangle$, predicatul fiind de mișcare iar calificatorul C de tipul (LA), (DE LA), (PÎNĂ LA), (PENTRU) etc.

Acest caz particular de frază poartă numele de *comandă*.

V. $\langle DACĂ \rangle \langle condiție \rangle \langle ATUNCI \rangle \langle comandă_1 \rangle \langle ALTFEL \rangle \langle comandă_2 \rangle$
în care $\langle condiție \rangle$ poate fi orice frază de tip I sau asocieri de fraze de tip I legate prin operatori logici, $\langle comandă_1 \rangle$ și $\langle comandă_2 \rangle$ fiind două fraze distincte de tipul IV.

Acest caz particular de frază poartă numele de *comandă condiționată*.

Observații. 1° Uneori, partea „ $\langle ALTFEL \rangle \langle comandă_2 \rangle$ ” din tipul de frază V poate să lipsească. În acest caz, dacă se îndeplinește $\langle condiție \rangle$ se execută $\langle comandă_1 \rangle$, dacă nu se îndeplinește $\langle condiție \rangle$ se trece la următoarea frază din text.

2° Dacă în cazurile IV și V predicatul P și noțiunile N sint de tip intern adică se referă la manipulări de fraze și noțiuni ale limbajului, atunci avem de-a face cu *comenzi interne* sau *instrucțiuni de metalimbaj*. Pentru a le diferenția de predicatele și noțiunile obișnuite ele primesc un asterisc (*) după ultima paranteză (din dreapta) care delimită entitatea, conform regulei de notație 4) menționată mai sus.

VI. Un grup de fraze alcătuiesc un text. Textele sint indexate și poartă denumirea de entități-index.

Utilizând notatiile BNF, în care pentru definiție se utilizează semnul „ $::=$ ” iar „ $|$ ” este un *metaconector* având rolul particulei de legătură „sau” [64], producțiile limbajului $L(G)$ se pot scrie :

a. $\langle Text \rangle \langle N \rangle$	$::= \langle N \rangle \langle Grup\ de\ fraze \rangle^*$
b. $\langle Grup\ de\ fraze \rangle$	$::= \langle Frază \rangle \langle Frază \rangle \langle Grup\ de\ fraze \rangle$
c. $\langle Frază \rangle$	$::= \langle Noțiune \rangle \langle Predicat \rangle \langle Noțiune \rangle $ $\langle Noțiune \rangle \langle Predicat \rangle \langle Noțiune \rangle$ $\langle Grup\ ON \rangle \langle Noțiune \rangle \langle Grup\ ON \rangle$ $\langle Predicat \rangle \langle Noțiune \rangle \langle Comandă \rangle $ $\langle Comandă\ condiționată \rangle$
d. $\langle Grup\ ON \rangle$	$::= \langle O \rangle \langle Noțiune \rangle \langle O \rangle \langle Noțiune \rangle$ $\langle Grup\ ON \rangle$
e. $\langle Noțiune \rangle$	$::= \langle N \rangle \langle N \rangle \langle Grup\ CN \rangle$
f. $\langle Grup\ CN \rangle$	$::= \langle Calificator \rangle \langle N \rangle \langle Calificator \rangle \langle N \rangle$ $\langle Grup\ CN \rangle$
g. $\langle Calificator \rangle$	$::= \langle O \rangle \langle C \rangle \langle Calificator \rangle$
h. $\langle Predicat \rangle$	$::= \langle P \rangle \langle NU \rangle \langle P \rangle$
i. $\langle Comandă \rangle$	$::= \langle SE \rangle \langle Predicat \rangle \langle Noțiune \rangle \langle SE \rangle$ $\langle Predicat \rangle \langle Noțiune \rangle \langle Grup\ ON \rangle$
j. $\langle Comandă\ condiționată \rangle$	$::= \langle DACĂ \rangle \langle Condiție \rangle \langle ATUNCI \rangle$ $\langle Comandă \rangle \langle DACĂ \rangle \langle Condiție \rangle \langle ATUNCI \rangle$ $\langle Comandă_1 \rangle \langle ALTFEL \rangle \langle Comandă_2 \rangle$
k. $\langle Condiție \rangle$	$::= \langle Noțiune \rangle \langle Predicat \rangle \langle Noțiune \rangle $ $\langle Noțiune \rangle \langle Predicat \rangle \langle Noțiune \rangle \langle Grup\ ONPN \rangle$
l. $\langle Grup\ ONPN \rangle$	$::= \langle O \rangle \langle Noțiune \rangle \langle Predicat \rangle \langle Noțiune \rangle \langle O \rangle$ $\langle Noțiune \rangle \langle Predicat \rangle \langle Noțiune \rangle \langle Grup\ ONPN \rangle$

*^o Pentru a facilita înțelegerea, în exemplele care urmează $\langle \underline{N} \rangle$ – titlul textului – va fi scris în rind separat și subliniat.

3.3. Predicate, funcții predicative și teoreme

Să considerăm în continuare fraza ca o funcție $f(e_1, e_2, \dots, e_n)$ de entitățile din care este alcătuită și să ne ocupăm de o clasă particulară a acestor funcții $e_p(e_1, e_2, \dots, e_n)$, în care e_p , este un predicat anume, avind o semnificație precisă $e_p \in \mathfrak{S}$. Convenim să numim aceste funcții particulare, drept *funcții predicative*.

În cadrul modelului INTELEG* destinat unor aplicații de inteligență artificială în economie, numărul predicatelor e_p admise — și care definesc tot atîtea funcții predicative — va fi destul de restrîns (lista tuturor acestor predicate împreună cu semnificația lor este dată în tabela 1).

Între aceste predicate există relații care permit transformarea unor funcții predicative în alte funcții predicative fără a le schimba înțelesul (se păstrează valoarea lor semantică) precum și trecerea de la o funcție predicativă la o funcție nepredicativă (fragment de frază fără predicat). În demonstrarea teoremelor care urmează se vor utiliza notațiile cunoscute precum și semnificațiile predicatelor trecute în tabela 1.

Teorema 1. *Propoziția $p_1 ::= \langle N_p \rangle$ [ESTE (CAZ PARTICULAR AL)] $\langle N_p \rangle$ este echivalentă cu propoziția $p_2 ::= \langle N_G \rangle$ [ESTE (ÎN ACEST CAZ)] $\langle N_p \rangle$, în care am notat cu: $\langle N_p \rangle =$ noțiunea — caz particular al noțiunii mai generale $\langle N_G \rangle$; $\langle N_G \rangle =$ noțiune care include $\langle N_p \rangle$ ca pe un caz particular, și se obțin una din cealaltă prin inversarea noțiunii subiect cu noțiunea complement și schimbarea predicatului.*

Demonstrație. p_1 și p_2 exprimă un anumit raport între două noțiuni $\langle N_p \rangle$ și $\langle N_G \rangle$, ceea din urmă fiind mai generală și incluzînd-o pe prima. Să notăm acest raport N_p/N_G . Funcția $\varphi(N_p/N_G)$ se notează relațional în două moduri diferite: fie utilizînd predicatul [ESTE (CAZ PARTICULAR AL)] și atunci $\langle N_G \rangle$ se plasează înaintea iar $\langle N_p \rangle$ se plasează după predicat, fie utilizînd predicatul [ESTE (ÎN ACEST CAZ)] și atunci poziția celor două noțiuni $\langle N_p \rangle$ și $\langle N_G \rangle$ se inversează. Dată fiind identitatea semantică a celor două propoziții p_1 și p_2 , echivalența lor este demonstrată.

Teorema 2. *Propoziția $p_a ::= \langle N_a \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_1 \rangle$ (și) $\langle N_2 \rangle$ (și) ... $\langle N_n \rangle$ este echivalentă cu grupul de propoziții:*

$$p_1 ::= \langle N_1 \rangle \text{ [FACE PARTE DIN]} \langle N_a \rangle$$

$$p_2 ::= \langle N_2 \rangle \text{ [FACE PARTE DIN]} \langle N_a \rangle$$

$$\dots$$

$$p_n ::= \langle N_n \rangle \text{ [FACE PARTE DIN]} \langle N_a \rangle$$

În care am notat cu $\langle N_a \rangle$ — noțiunea ansamblu, $\langle N_1 \rangle$, $\langle N_2 \rangle$, ..., $\langle N_n \rangle$ — noțiunile elemente ale ansamblului, și se obțin una din cealaltă prin descompunere (respectiv, compunere), inversarea noțiunii subiect cu noțiunea complement și schimbarea predicatului.

Demonstrație: Atât p_a cît și ansamblul de propoziții (p_1, p_2, \dots, p_n) exprimă același raport între o noțiune-ansamblu $\langle N_a \rangle$ și componente sale $\langle N_1 \rangle, \langle N_2 \rangle, \dots, \langle N_n \rangle$. Să notăm acest raport cu $N_a/(N_1, N_2, \dots, N_n)$. Funcția $\Psi(N_a/(N_1, N_2, \dots, N_n))$ se notează relațional în două moduri diferite: fie utilizînd predicatul [SE COMPUNE DIN (ANSAMBLU)] și atunci $\langle N_a \rangle$ se plasează înaintea predicatului avind rol de subiect în timp ce toate celealte $\langle N_i \rangle$, $i = 1, \dots, n$ legate prin (și) se plasează după predicat, avind rol de complement, fie utilizînd predicatul [FACE PARTE

* INTELIGENȚĂ ECONOMICĂ.

Tabelă 1
LISTA PREDICATELOR ȘI SEMNIFICAȚIA LOR

Nr. cert.	Predicat	Semnificație
1.	ESTE (CAZ PARTICULAR AL)	Subiectul este cazul particular al noțiunii exprimată prin complement, ceea ce înseamnă că toate frazele în care apare noțiunea complement sunt valabile și pentru noțiunea subiect și se obțin din primele înlocuind noțiunea complement cu noțiune subiect. Atunci cînd se efectuează această înlocuire trebuie, de regulă, precizat și înțelesul celorlalte noțiuni generale $\langle N_c \rangle$ care apar în aceste fraze, cu ajutorul unor propoziții suplimentare de forma $\langle N_c \rangle$ [ESTE (ÎN ACEST CAZ)] $\langle N_p \rangle$, unde am notat cu $\langle N_p \rangle$ valoarea particulară a noțiunii $\langle N_c \rangle$ în cazul respectiv.
2.	ESTE (ÎN ACEST CAZ)	Subiectul este o noțiune mai generală care în cazul de față (textul curent) are semnificația noțiunii complement.
3.	SE COMPUNE DIN (ANSAMBLU)	Subiectul este un ansamblu format din elemente specificate printr-un sir de complemente legate între ele prin operatorul (și). Ordinea de enumerare nu contează.
4.	FACE PARTE DIN	Subiectul este un element al ansamblului exprimat prin noțiunea complement.
5.	SE COMPUNE DIN (SUCCESIUNE)	Subiectul este un element compus dintr-o succesiune de alte elemente, acestea din urmă fiind complemente legate între ele prin operatorul (URMĂDE). Se aplică, de exemplu, la un proces pentru realizarea căruia trebuie parcurs, în ordine, mai multe faze.
6.	PRECEDE	Subiectul precede (în timp sau în spațiu) complementul.
7.	SUCCEDE	Subiectul succede (în timp sau în spațiu) complementului.
8.	COORDONEAZA	Subiectul este ierarhic superior și coordonează noțiunile exprimate printr-o serie de complemente legate între ele prin operatorul (și). Orice comandă a subiectului se execută de către noțiunile complement.
9.	ESTE SUBORDONAT LUI	Subiectul este ierarhic inferior și execută orice comandă a noțiunii complement.

Nr. crt.	P r e d i c a t	S e m n i f i c a t i e
10.	SE REFERA LA	Subiectul nu este nici cazul particular nici generalizarea complementului dar are o oarecare legătură cu acesta din urmă (intersecția mulțimilor caracteristicilor celor două noțiuni — subiect și complement — este nevidă).
11.	VERB DE MIȘCARE*)	Subiectul execută o acțiune explicitată prin complement.
12.	ARE	Subiectul are în componență să anumite noțiuni precizate în complementele care îl succed. Acestea din urmă însă nu caracterizează total subiectul.
13.	MERGE LA	Dacă subiectul este 〈SE〉 , atunci predicatul indică o comandă (de obicei internă, arătând prin complement fraza cu care se continuă sau textul care urmează). Dacă subiectul nu este 〈SE〉 atunci predicatul intră în categoria verbelor de mișcare.
14.	ESTE DERIVAT DIN	Dacă predicatul este intern, arată că subiectul (substantiv) este derivat din complement (verb de mișcare).
15.	ESTE FUNCTIE DE	Exprimă o relație funcțională între complemente, relație de forma $y = f(x)$ care definește subiectul y . Explicitarea acestei relații va fi efectuată cu ocazia particularizării lui $f(x)$.
16.	ÎNLOCUIESTE	Dacă predicatul este intern, arată că entitatea-complement situată înaintea calificatorului (cu) se înlocuiește cu entitatea complement situată după acest calificator. Înlocuirea are loc oriunde prima entitate este întâlnită în cadrul paragrafului marcat cu (DIN) sau (AL/A) <i>Exemplu</i> : 〈SE〉 [INLOCUIESTE]* 〈OPERATIE〉 (DIN) 〈SUBPROGRAM〉 (CU) 〈PRIMA OPERATIE〉 (AL/A) 〈PROCEDURA〉 are următoarea semnificație : în cadrul paragrafului 〈SUBPROGRAM〉 se caută 〈OPERATIE〉 și oriunde este întâlnită se înlocuiește cu 〈PRIMA OPERATIE〉 care aparține paragrafului 〈PROCEDURA〉 .
17.	EXECUTA	Dacă predicatul este intern, arată că trebuie parcuse secvențial toate frazele legate la entitatea-complement, după care se întoarce și se execută fraza care urmează.

*) Acesta nu este propriu-zis un predicat, ci o categorie de predicate în care intră toate verbele de mișcare nespecificate în acest tabel.

Nr.
crt.

Predicat

Semnificație

18. INREGISTREAZA

Exemplu : <SE> [EXECUTA] * <SUBPROGRAM>; se parcurg toate frazele legate la entitatea <SUBPROGRAM>; se revine apoi și se citește fraza care urmează acelora de mai sus. Dacă predicatul nu este intern, execuția se referă la alte proceduri care nu sunt efectuate automat de calculator.

Dacă predicatul este intern, se generează fraza : <X> [SE COMPUNE DIN (SUCCESSIONE)] <X₁> (URMAT DE) <X₂> ... în care <X> este entitatea complement generală situată după calificatorul (LA) iar <X₁>, <X₂>, ... sunt valorile ei particulare, plasate înaintea acestui calificator.

Observație. Dacă fraza a fost deja generată, cu un sir de valori, și predicatul [INREGISTREAZA] * indică altă serie de valori, acestea din urmă se adaugă la valorile existente prin operatorul (URMAT DE)

Exemplu : În BDR există fraza :

<PROCEDURA EXECUTATA> [SE COMPUNE DIN (SUCCESSIONE) <OPERATIE 1> (URMAT DE) <OPERATIE 2>]

Se citește fraza :

<SE> [ÎNREGISTREAZA] * <OPERATIE 3> (LA) <PROCEDURA EXECUTATA> ceea ce are ca efect completarea primei fraze în sensul : <PROCEDURA EXECUTATA> [SE COMPUNE DIN (SUCCESSIONE)] <OPERATIE 1> (URMAT DE) <OPERATIE 2> (URMAT DE) <OPERATIE 3>.

Dacă predicatul este intern, atunci se revine la paragraful de la care s-a plecat prin comanda anterioară [EXECUTA] *

Exemplu :

1. <SE> [EXECUTA] * <SUBPROGRAM>	<u><PROGRAM PRINCIPAL></u>
2. <DACA> ...	

1. ...	<u><SUBPROGRAM></u>
2. ...	

Nr. crt.	Predicat	Semnificație
	3. <DACA> <condiție> <ATUNCI> <SE> [INTOARCE LA]* <PROGRAM PRINCIPAL> 4. ...	
20. ESTE EGAL CU		Se execută prima comandă din <PROGRAM PRINCIPAL>. Se trece la <SUBPROGRAM>. Dacă <condiție> este satisfăcută atunci se întoarce la <PROGRAM PRINCIPAL> și citește fraza 2.
21. OPRESTE		Dacă predicatul este intern și subiectul este precedat de <DACA> atunci se compară subiectul cu complementul și la egalitatea acestora se execută comanda care urmează (vezi comanda <i>conditionată</i>).
22. ACTUALIZEAZA		Dacă predicatul este intern acesta indică oprirea acțiunilor execuțate în urma comenzi precedente EXEC ... PROGRAM a utilizatorului.
23. STABILESTE		Dacă predicatul este intern, atunci entitatea — complement <C> situată înaintea calificatorului (IN) înlocuiește complementul <C> al unei fraze existente, de tipul :
	<X> [ESTE (IN ACEST CAZ)] <C>	
		în care <X> este entitatea-complement situată după calificatorul (IN) în fraza noastră.
		<i>Exemplu</i> : <SE> [ACTUALIZEAZA] * <MODUL SIM1B> (IN) <OPERATIA URMATOARE>. Se caută fraza în care <OPERATIA URMATOARE> este subiect, predicatul fiind [ESTE (IN ACEST CAZ)]. Să presupunem că se găsește :
		<OPERATIA URMATOARE> [ESTE (IN ACEST CAZ)] <PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE>. Se înlocuiește complementul conform regulii enunțate, fraza de mai sus devenind în cele din urmă <OPERATIA URMATOARE> [ESTE (IN ACEST CAZ)] <MODUL SIM1B>
		Dacă predicatul este intern, atunci fraza <SE> [STABILESTE]* <X> (LA) <Y> se traduce prin acordarea lui <X> a valorii lui <Y>, cu alte cuvinte dacă <Y> [ESTE (IN ACEST CAZ)] <V>

Nr. crt.	Predicat	Semnificație
24.	FIXEAZA	<p>atunci $\langle X \rangle$ devine: $\langle X \rangle$ [ESTE (IN ACEST CAZ)] $\langle V \rangle$, orice valoare ar fi avut inainte. Dacă predicatul este intern, el semnifică accordarea unei valori complementului (de către utilizator).</p> <p><i>Exemplu:</i> $\langle \text{SE} \rangle$ [FIXEAZA]* $\langle \text{TST} \rangle$ (AL/A) $\langle \text{COMANDA} \rangle$ generează următorul dialog:</p> <p>c : COMANDA ? u : COM010 c : TST u : 12 c : COMANDA ? u : COM011 c : TST u : 10 etc.</p>
25.	AFISEAZA	<p>În urma dialogului se generează următoarele fraze: $\langle \text{TST} \rangle$ (AL/A) $\langle \text{COM010} \rangle$ [ESTE (IN ACEST CAZ)] $\langle 12 \rangle$ $\langle \text{TST} \rangle$ (AL/A) $\langle \text{COM011} \rangle$ [ESTE (IN ACEST CAZ)] $\langle 11 \rangle$ etc. Dacă predicatul este intern, el indică afișarea (pe ecranul terminalului) a frazelor care compun paragraful menționat.</p> <p><i>Exemplu:</i> $\langle \text{SE} \rangle$ [AFISEAZA]* $\langle \text{DEFINITIE} \rangle$ (AL/A) $\langle \text{MODUL SIM1B} \rangle$. Urmează listarea frazelor de definiție ale modulului SIM1B</p> <p><i>Observație.</i> Dacă complementul este menționat între două semne apostrof ('...') atunci se afișează chiar textul dintre aceste semne.</p> <p><i>Exemplu:</i> $\langle \text{SE} \rangle$ [AFISEAZĂ]* $\langle 'NOUA SELECTIE ?' \rangle$ duce la apariția pe ecran a textului:</p> <p style="text-align: center;">NOUA SELECTIE ?</p>
26.	ASTEAPTA RASPUNS	<p>Dacă predicatul este intern, el provoacă așteptarea sistemului de calcul pînă cînd utilizatorul va furniza, la terminal, răspunsul indicat. Este cazul ultimului exemplu de la p. 25 cînd a fost formulată o întrebare pentru utilizator.</p>

DIN] și atunci se formează n propoziții, fiecare avind $\langle N_i \rangle$ drept subiect și $\langle N_a \rangle$ drept complement.

Corolar. Predicatul [ARE] în cadrul propoziției $p'_a :: = \langle N_a \rangle$ [ARE] $\langle N_1 \rangle$ (SI) $\langle N_2 \rangle$ (SI) ... $\langle N_n \rangle$ păstrează aceeași proprietate demonstrată mai sus pentru predicatul [SE COMPUNE DIN (ANSAMBLU)] în cadrul propoziției p_a , propoziția p'_a fiind deci echivalentă cu grupul de propoziții p_1, p_2, \dots, p_n și putindu-se transforma în acesta respectind regulile arătate pentru p_a . Demonstrația se bazează pe faptul că semnificația predicatului [ARE] (vezi tabela 1) arată tot un raport de la ansamblu la element caracteristic al acestuia, de forma $\Psi(N_a/N_1, N_2, \dots, N_n)$ chiar dacă, în acest caz, N_1, N_2, \dots, N_n nu epuizează complet definiția ansamblului.

Teorema 3. Propoziția $p_a :: = \langle N_a \rangle$ [SE COMPUNE DIN (SUCCESIUNE)] $\langle N_1 \rangle$ (URMAT DE) $\langle N_2 \rangle$ (URMAT DE) ... $\langle N_n \rangle$ este echivalentă cu grupul de propoziții :

$$p_1 :: = \langle N_1 \rangle \text{ [PRECEDE] } \langle N_2 \rangle$$

$$p_2 :: = \langle N_2 \rangle \text{ [PRECEDE] } \langle N_3 \rangle$$

$$\dots$$

$$p_{n-1} :: = \langle N_{n-1} \rangle \text{ [PRECEDE] } \langle N_n \rangle$$

$$p_n :: = \langle N_n \rangle \text{ [SE COMPUNE DIN (ANSAMBLU)] } \langle N_1 \rangle \text{ (SI) } \langle N_2 \rangle \text{ (SI) } \dots \langle N_n \rangle$$

format cu ajutorul predicatelor [PRECEDE] și [SE COMPUNE DIN (ANSAMBLU)] sau cu grupul de propoziții :

$$p'_1 :: = \langle N_2 \rangle \text{ [SUCCÈDE] } \langle N_1 \rangle$$

$$p'_2 :: = \langle N_3 \rangle \text{ [SUCCÈDE] } \langle N_2 \rangle$$

$$p'_{n-1} :: = \langle N_n \rangle \text{ [SUCCÈDE] } \langle N_{n-1} \rangle$$

$$p'_n :: = \langle N_n \rangle \text{ [SE COMPUNE DIN (ANSAMBLU)] } \langle N_1 \rangle \text{ (SI) } \langle N_2 \rangle \text{ (SI) } \dots \langle N_n \rangle$$

în care intervin predicatele [SUCCÈDE] și [SE COMPUNE DIN (ANSAMBLU)]. O propoziție (p'_i) , $i = 1, \dots, n-1$ se obține din propoziția (p_i) , $i = 1, \dots, n-1$, echivalentă, inversând rolul noțiunilor subiect și complement (schimbând pe $\langle N_i \rangle$ cu $\langle N_{i-1} \rangle$ și vice-versă) și modificând predicatul. Pentru a obține o propoziție (p_i) , $i = 1, \dots, n-1$ din echivalența sa (p'_i) , $i = 1, \dots, n-1$, se procedează invers ca în cazul precedent.

Demonstrație. p_a , grupul $(p_1, p_2, \dots, p_{n-1}, p_n)$ și grupul $(p'_1, p'_2, \dots, p'_{n-1}, p'_n)$ exprimă același lucru și anume succesiunea unor elemente $\langle N_i \rangle$, $i = 1, \dots, n$ care, la un loc, alcătuiesc ansamblul $\langle N_a \rangle$. Funcția $\Psi'(N_a/(N_1, N_2, \dots, N_n))$ care exprimă acest raport se notează în modurile arătate mai sus, prin propoziția p_a sau grupul propozițiilor (p_i) , $i = 1, \dots, n$, sau grupul (p'_i) , $i = 1, \dots, n$. Analog teoremei 1, raportul de succesiune între noțiunile $N_{i-1}/, N_i$ poate fi exprimat în două feluri : fie așezând $\langle N_{i-1} \rangle$ înaintea predicatului [PRECEDE] și noțiunea $\langle N_i \rangle$ după acest predicat, fie așezând $\langle N_i \rangle$ înaintea predicatului [SUCCÈDE] și pe $\langle N_{i-1} \rangle$ după predicatul ultim menționat mai sus.

Teorema 4. Propoziția $p_c ::= \langle N_c \rangle$ [COORDONEAZA] $\langle N_1 \rangle$ (și) $\langle N_2 \rangle$ (și) ... $\langle N_n \rangle$ este echivalentă cu grupul de propoziții:

$p_1 ::= \langle N_1 \rangle$ [ESTE SUBORDONAT LUI] $\langle N_c \rangle$

$p_2 ::= \langle N_2 \rangle$ [ESTE SUBORDONAT LUI] $\langle N_c \rangle$

$p_n ::= \langle N_n \rangle$ [ESTE SUBORDONAT LUI] $\langle N_c \rangle$

în care am notat cu: $\langle N_c \rangle$ – elementul coordonator al acțiunilor elementelor $\langle N_1 \rangle, \dots, \langle N_n \rangle$; $\langle N_1 \rangle, \langle N_2 \rangle, \langle N_3 \rangle, \dots, \langle N_n \rangle$ – elemente subordonate lui $\langle N_c \rangle$ și se obțin una din cealaltă prin descompunere (respectiv, compunere), inversarea noțiunii subiect cu noțiunea complement și schimbarea predicatului.

Demonstrație. p_c și grupul de propoziții (p_1, p_2, \dots, p_n) exprimă același raport de subordonare a acțiunilor lui $\langle N_1 \rangle, \langle N_2 \rangle, \dots, \langle N_n \rangle$ față de $\langle N_c \rangle$, care se notează $N_c/_{\text{I}}(N_1, N_2, \dots, N_n)$. Funcția $\pi(N_c/_{\text{I}}(N_1, N_2, \dots, N_n))$ se exprimă deci în două moduri diferite: primul, utilizând predicatul [COORDONEAZA] și atunci $\langle N_c \rangle$ este situat înaintea predicatului iar toate noțiunile $\langle N_i \rangle$, $i = 1, \dots, n$, subordonate lui sunt plasate după predicat (legate între ele prin [și]); al doilea, utilizând predicatul [ESTE SUBORDONAT LUI] și formind n propoziții în care $\langle N_i \rangle$, $i = 1, \dots, n$, au rol de subiect iar $\langle N_c \rangle$ are rolul de complement.

Teorema 5. Fie șirul de propoziții:

$p_1 ::= \langle N_1 \rangle$ [ESTE (CAZ PARTICULAR AL)] $\langle N_2 \rangle$

$p_2 ::= \langle N_2 \rangle$ [ESTE (CAZ PARTICULAR AL)] $\langle N_3 \rangle$

$p_{n-1} ::= \langle N_{n-1} \rangle$ [ESTE (CAZ PARTICULAR AL)] $\langle N_n \rangle$

Orice funcție propozițională a lui $\langle N_n \rangle$, fie aceasta $\Phi(N_n)$ este valabilă și pentru $\langle N_{n-1} \rangle, \dots, \langle N_2 \rangle, \langle N_1 \rangle$, cu alte cuvinte $\exists \Phi(N_{n-1}), \dots, \Phi(N_2), \Phi(N_1)$ care se obțin din prima înlocuind $\langle N_n \rangle$ succesiv cu $\langle N_{n-1} \rangle, \dots, \langle N_2 \rangle, \langle N_1 \rangle$ și precizând celelalte noțiuni $\langle N_c \rangle$ care intervin.

Demonstrație. Conform definiției predicatului [ESTE (CAZ PARTICULAR AL)] (vezi tabela 1) $\Phi(N_{n-1})$ se obține, având în vedere p_{n-1} și înlocuind în $\Phi(N_n)$ pe $\langle N_n \rangle$ cu $\langle N_{n-1} \rangle$. În mod analog se obține $\Phi(N_{n-2})$ din $\Phi(N_{n-1})$ având în vedere propoziția p_{n-2} și.a.m.d., $\Phi(N_1)$ se obține din $\Phi(N_2)$ și propoziția p_1 .

Teorema 6. Fie șirul de propoziții:

$p^a ::= \langle N \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_1 \rangle$ (și) $\langle N_2 \rangle$ (și) ...

$p_1^b ::= \langle N_1 \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_{11} \rangle$ (și) $\langle N_{12} \rangle$ (și) ...

$p_2^b ::= \langle N_2 \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_{21} \rangle$ (și) $\langle N_{22} \rangle$ (și) ...

$p_{11}^c ::= \langle N_{11} \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_{111} \rangle$ (și) $\langle N_{112} \rangle$ (și) ...

$p_{12}^c ::= \langle N_{12} \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_{121} \rangle$ (și) $\langle N_{122} \rangle$ (și) ...

$p_{21}^c ::= \langle N_{21} \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_{211} \rangle$ (și) $\langle N_{212} \rangle$ (și) ...

$p_{22}^c ::= \langle N_{22} \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_{221} \rangle$ (și) $\langle N_{222} \rangle$ (și) ...

etc.

Tinind seama de cele de mai sus, noțiunea $\langle N_{ijk} \rangle$ este complet definită prin fraza nepredicativă :

$$f_{ijk} ::= \langle N_{ijk} \rangle \text{ (AL/A)} \langle N_{ij} \rangle \text{ (AL/A)} \langle N_i \rangle \text{ (AL/A)} \langle N \rangle$$

care poate fi folosită ca noțiune pe post de subiect sau complement în oricare altă propoziție.

Demonstrație. Calificatorul (AL/A) (vezi tabela 2) așezat în față unei noțiuni $\langle N_2 \rangle$ și după o noțiune $\langle N_1 \rangle$ are rolul de a indica apartenența lui

Tabela 2

LISTA CALIFICATORILOR ȘI SEMNIFICAȚIILE LOR

Nr. crt.	Calificator	Semnificație
1.	AL/A	Noțiunea dinaintea calificatorului aparține noțiunii plasate după acest calificator, ca parte a unui întreg, element al unui ansamblu, fază a unui proces, organierarhie subordonat etc.
2.	PL	Indică pluralul noțiunii care urmează.
3.	FIECARE	Indică faptul că predicatul se referă la fiecare caz particular al noțiunii care urmează.
4.	UNII/UNELE	Indică faptul că predicatul se referă numai la unele cazuri particulare ale noțiunii care urmează și pe care o califică.
5.	UN/O	Idem, ca în cazul precedent, cu deosebirea că indică un singur caz particular.

Restul calificatorilor, nespecificați în acest tabel, se interpretează în funcție de predicatul frazei respective (vezi tabela 1).

$\langle N_1 \rangle$ la $\langle N_2 \rangle$ ca parte componentă la întreg. Cu alte cuvinte $\langle N_1 \rangle$ (AL/A) $\langle N_2 \rangle$ are aceeași semnificație ca propoziția $\langle N_1 \rangle$ [FACE PARTE DIN] $\langle N_2 \rangle$ care, conform teoremei 2 am văzut că este echivalentă cu propoziția $\langle N_2 \rangle$ [SE COMPUNE DIN (ANSAMBLU)] $\langle N_1 \rangle$ (SI) ... Înlocuind pe $\langle N_2 \rangle$ și $\langle N_1 \rangle$ cu $\langle N_{ij} \rangle$, respectiv $\langle N_{ijk} \rangle$ obținem echivalența :

$$\langle N_{ijk} \rangle \text{ (AL/A)} \langle N_{ij} \rangle \equiv \langle N_{ij} \rangle \text{ [SE COMPUNE DIN (ANSAMBLU)]} \langle N_{ijk} \rangle \text{ (SI)} \dots$$

ultima propoziție fiind de tipul $p_{i1}^e, p_{i2}^e, \dots, p_{21}^e, p_{22}^e, \dots$ date prin ipoteză.

În mod analog se demonstrează că :

$$\langle N_{ij} \rangle \text{ (AL/A)} \langle N_i \rangle \equiv \langle N_i \rangle \text{ [SE COMPUNE DIN (ANSAMBLU)]} \langle N_{ij} \rangle \text{ (SI)} \dots$$

aceasta din urmă fiind de tipul p_1^b, p_2^b, \dots în timp ce :

$$\langle N_i \rangle \text{ (AL/A)} \langle N \rangle \equiv \langle N \rangle \text{ [SE COMPUNE DIN (ANSAMBLU)]} \langle N_i \rangle \text{ (SI)} \dots$$

rezultă luând în considerare prima propoziție, p^a .

Observație. Teorema de mai sus este valabilă și pentru funcțiile predicative :

$$[\text{SE COMPUNE DIN (SUCCESSIONE)}] f(N, N_i, N_{ij}, N_{ijk}, \dots) \text{ și } [\text{COORDONEAZA}] f'(N, N_i, N_{ij}, N_{ijk}, \dots)$$

numai că, în aceste cazuri, fraza nepredicativă $\langle N_{ijk} \rangle$ (AL/A) $\langle N_{ij} \rangle$ (AL/A) ... are înțelesul de *frază a unui proces* (pentru funcția f) și, respectiv de *organ subordonat comenzilor* (pentru funcția f').

★

Teoremelor de mai sus le corespunde, din punct de vedere formal, un număr suplimentar de producții care se adaugă la setul principal $a-l$, de mai sus, și contribuie la îmbogățirea limbajului. De asemenea, noțiunile și predicatele interne dă naștere, atunci cînd sunt utilizate în fraze de tip *comandă* și *comandă condițională*, la o altă serie de producții prin care se obțin noi fraze și texte. Nu vom insista asupra acestor formalisme deoarece considerăm că ele ar încărea în mod abuziv acest capitol neaducind o contribuție esențială la înțelegerea noțiunilor care urmează.

3.4. Definiția bazei de date relaționale (BDR)

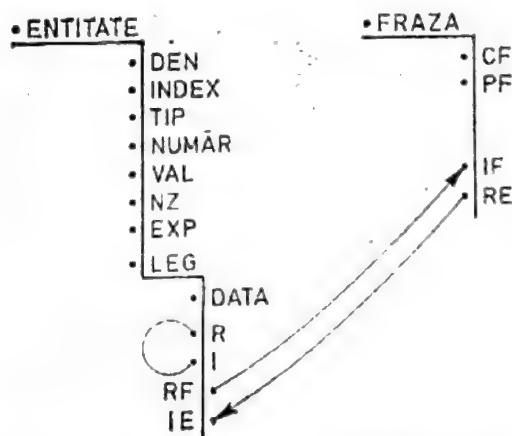
În vederea stocării și manipulării entităților $e \in E$, precum și a frazelor $f \in F$ în cadrul limbajului $L(G)$ definit mai sus, avem nevoie de o bază de date.

O bază de date este definită matematic printr-un ansamblu de entități și de relații $B(E, R)$, [49]. Dacă prin entități înțelegem ceea ce am definit anterior prin acest termen pentru $L(G)$, iar în relații includem toate relațiile \mathfrak{R}^2 , \mathfrak{R}^3 , etc. între entități, cu alte cuvinte frazele $f \in F$ construite conform gramaticii G definită anterior, *baza de date* $B(E, R)$ nu va fi altceva decât *suportul informatic al limbajului* $L(G)$.

Baza de date $B(E, R)$ se numește *relațională* deoarece nu conține decât entități, legături între entități și procedee de manipulare a sirurilor de entități (frazelor) în scopul descoperirii unor noi relații între acestea.

Instrumentul informatic prin care se realizează $B(E, R)$ trebuie să fie capabil să *comunice* cu utilizatorul — să pună întrebări și să înregistreze răspunsurile acestuia, să sesizeze și să comunice erorile formale precum și pe cele logice, să prelucreze informațiile primite conform unor proceduri generale sau particulare existente la momentul respectiv în biblioteca de proceduri a sistemului. Cu alte cuvinte, este nevoie de un sistem *interactiv* care să conlucreze cu utilizatorul în scopul găsirii celei mai bune soluții la problema formulată de utilizator.

Fig. 2. — Structura bazei de date relaționale a modelului INTELEC.



Pentru a satisface necesitățile de manipulare a limbajului $L(G)$ definit anterior am propus următoarea structură $S(E, F)$ a bazei de date relaționale (vezi fig. 2).

a. *entități de nivel 1 : ENTITATE, FRAZA,
entități de nivel 2 : LEG (subentitate a ENTITATE).*

b. *Legături :*

De la o ENTITATE la altă ENTITATE prin LEG (legătură de nivel 2).

De la o ENTITATE la o FRAZĂ prin LEG (legătură între nivel 2 și nivel 1).

De la o FRAZĂ la o ENTITATE prin LEG (legătură între nivel 1 și nivel 2).

c. *Caracteristici :*

ENTITATE — entitate, cheie de regăsire numerică admisă între limitele 1 și n , cuprinde următoarele caracteristici :

— DEN = denumire — sir de caractere, vector de lungime variabilă format din punerea cap la cap a cuvintelor de lungime variabilă care exprimă entitatea ;

— INDEX = caracteristică de tip a (vezi definiția limbajului) — sir de caractere, vector de lungime variabilă format din punerea cap la cap a unor șiruri de lungime fixă (30 caractere), fiecare din ele conținind un „cuvint cheie” referitor la entitate ;

— TIP = caracteristică de tip b (vezi definiția limbajului) — sir de caractere de lungime egală cu 1, putind avea una din valorile N , P , C , O , K , Q referitoare la entitate ;

— NUMĂR = caracteristică de tip c (vezi definiția limbajului) — sir de caractere de lungime egală cu 1, putind avea una din valorile : I , R , E , care referă entitățea. Lipsa oricărei valori ne arată că nu avem de-a face cu o entitate numerică ;

— VAL = caracteristică numerică (vezi definiția limbajului), cuprinsă între valorile — 99 999 999 și + 99 999 999, necesară pentru memorarea cifrelor semnificative ale numărului ;

— NZ = caracteristică numerică (vezi definiția limbajului), cuprinsă între valorile 0 și 8, necesară pentru memorarea numărului de zecimale ;

— EXP = caracteristică numerică (vezi definiția limbajului), cuprinsă între valorile —63 și +64, necesară pentru memorarea valorii exponentului în baza 10.

FRAZA — entitate, cheie de regăsire numerică admisă între limitele 1 și m , cuprinde următoarele caracteristici :

— CF = caracteristică sir de caractere de lungime egală cu 1, putind avea una din valorile : D , P , G , A , acestea având semnificațiile menționat cu ocazia definiției limbajului ;

— PF = caracteristică numerică, cuprinsă între 0 și 100, servind la memorarea probabilității de adevăr cu care poate fi creditată fraza (vezi paragraful referitor la definiția limbajului) ;

— IF = caracteristică de tip inel, care, pereche cu caracteristica de tip referință RF, servește la legarea entităților în cadrul aceleiași fraze ;

— RE = caracteristică de tip referință, care, pereche cu caracteristica de tip inel IE, servește la legarea frazelor din cadrul aceluiași text al cărui titlu este exprimat printr-o entitate).

LEG — subentitate a entității ENTITATE, cheie de regăsire numerică între limitele 1 și p , cuprinde următoarele caracteristici :

— DATA = caracteristică numerică, cuprinsă între limitele „aallzz” și „a'a' l'l' z'z'” servind la memorarea unei date (an, lună, zi). Cifrele au următoarea semnificație : aa — primele două cifre sunt egale cu ultimele două cifre ale anului (exemplu : 82 înseamnă anul 1982); ll — următoarele două cifre exprimă numărul de ordine al lunii în cadrul anului

(exemplu : 10 înseamnă luna octombrie); zz — următoarele două cifre exprimă ziua calendaristică în cadrul lunii, exemplu : 28 înseamnă a 28-a zi din luna respectivă;

Exemplu : Numărul 820714 va reprezenta, prin urmare, data de 14 iulie 1982.

Data servește la compararea a două texte scrise în perioade diferite de timp sau la evaluarea mai precisă a semnificațiilor unor fraze (plasarea în timp a acțiunii exprimate de acestea) :

— R : caracteristică de tip *referință* care, pereche cu caracteristica de tip *inel I* servește la legarea între ele a două entități (relația de succesiune a două entități în cadrul frazei) ;

— I : caracteristică de tip *inel*, pereche cu caracteristica R de mai sus ;

— RF : caracteristică de tip *referință* care, pereche cu caracteristica de tip *inel IF* servește la legarea entităților în cadrul frazei ;

— RE : caracteristică de tip *inel* care, pereche cu caracteristica de tip *referință RE* servește la legarea frazelor din cadrul aceluiași text (al cărui titlu este exprimat printr-o entitate).

3.5. Principii de utilizare a bazei de date relaționale

Pentru utilizarea bazei de date relaționale definită mai sus se vor respecta următoarele *principii* :

1° Orice entitate se va înregistra o singură dată în BDR, cu toate cuvintele componente scrise în ordine, unul după altul.

2° Orice frază se va forma prin *legături de tip R—I* între entități. În același timp *toate entitățile componente vor fi legate la inelul IF* al fazei prin referințe de tip RF. Dacă entitatea care urmează la rind există deja în BDR atunci se va genera *numai o subentitate LEG* și se va proceda la crearea legăturilor R—I (cu subentitatea LEG a entității precedente) și RF—IF (cu fraza). Dacă entitatea care urmează la rind nu există în BDR se va proceda mai întii la înregistrarea acesteia, conform pct. 1° de mai sus, după care se vor crea legăturile.

3° O frază nu se va putea înscrie decât după crearea, în prealabil, a entității **FRAZA** respective.

4° La crearea entității **FRAZA** este obligatorie afilierea ei la o **ENTITATE**-text existentă sau pe care o putem crea în prealabil. Completarea caracteristicilor CF și PF este optională.

5° Entitatea-text la care urmează să fie legată fraza va avea completeate caracteristicile **DEN** și **eventual, INDEX** ; Se va genera o subentitate **LEG** și se va completa, obligatoriu, **DATA**. Este posibilă și legarea frazei la o subentitate **LEG** existentă (având deja fraze în subordine).

6° Entitățile componente ale unei fraze vor avea, obligatoriu, completeate caracteristicile **DEN, TIP** și, **eventual, INDEX și NUMAR**. Dacă **NUMAR** este completată atunci, obligatoriu, **VAL** va fi completată și de la caz la caz **NZ** și **EXP** vor primi și ele valori numerice.

Structura de mai sus, creată pentru baza de date relațională, poate fi oricând completată cu noi caracteristici, atât pentru entități cît și pentru fraze, în măsura în care acestea servesc unor scopuri urmărite de utilizator, fără a afecta în vreun fel informațiile existente.

De exemplu : se pot adăuga caracteristici pentru înregistrarea frecvenței de utilizare a textelor, frazelor și procedurilor în scopul unor studii statistice. În funcție de această frecvență unele texte și fraze neutilizate sau slab utilizate se pot elibera din baza de date relațională, eliberind memoria calculatorului și adaptând-o în permanență cerințelor practicii. Această posibilitate este similară, într-o oarecare măsură, cu aceea a creierului omenesc.

3.6. Modul de utilizare a bazei de date relaționale

Pentru exemplele care urmează și care demonstrează posibilitățile oferite de BDR în domeniul prelucrării informațiilor, vom utiliza notatiile introduse cu ocazia definirii limbajului INTELEC. Amintim aici că frazele vor fi reprezentate prin succesiuni de entități închise între semnele : „< „ și „>” pentru noțiuni, „[„ și „]” pentru predicate, „(„ și „)” pentru calificatori și operatori. Noțiunile și predicatele interne vor primi semnul „*” după ultima paranteză (din dreapta).

Notății utilizate în scheme. Entitățile și subentitățile sunt reprezentate în scheme prin linii arborescente iar legăturile prin săgeți care pornesc din referințe și ajung în inelele respective. Caracteristicile entităților sunt marcate prin puncte așezate de-o parte și de alta a liniilor verticale ce reprezintă entitățile.

Reprezentarea frazelor în structura BDR se va face prin scheme de arbori și legături, păstrând convențiile de mai sus.

a. Introducerea informațiilor în baza de date relațională

Exemplul 1 : Să presupunem, că dorim să introducem sau să adăugăm informații în legătură cu *utilizarea pachetului de programe SOFTPLAN* pentru conducerea operativă a producției în ramura construcțiilor de mașini*). Această operație se va desfășura sub forma unui *dialog* cu calculatorul**), în care acesta din urmă va indica pas cu pas ce informații se cer și forma sub care trebuie introduse pentru a realiza obiectivul propus de utilizator. Pentru a iniția dialogul, utilizatorul „cheamă” numele unui program monitor care supraveghează desfășurarea întregii operații. În cazul de față acest program poartă numele de „ACTUALIZARE”

Comenzi și răspunsuri la
consola terminalului de
teletransmisie :

u : EXEC ACTUALIZARE ?
c : ENTITATE ?

Note
explicative :

Calculatorul întreabă numele entității de care vor fi legate informațiile ce urmează a fi introduse în BDR.

*) Vezi descrierea acestui pachet de programe în cadrul capitolului 6.

**) În cadrul dialogului am notat cu „u” ceea ce comandă sau răspunde utilizatorul și cu „c” răspunsurile sau întrebările calculatorului.

Comenzi și răspunsuri la
consola terminalului de
teletransmisie:

Note
explicative:

**u : PROCEDURA DE UTILIZARE
SOFTPLAN**

**c : NU EXISTA. SE INTRODUC
(1 = DA ; 0 = NU) ?**

u : 1

**c : INTRODUS ÎN BDR.
ENTITATE 17.
INDEX ?**

u : SOFTPLAN

c : INDEX ?

u : UTILIZARE

c : INDEX ?

u : *

c : TIP ?

u : N

c : DATA ?

**u : 811015
FRAZA**

c : 31

**u : PROCEDURA DE UTILIZARE
SOFTPLAN ESTE (CAZ
PARTICULAR AL)
PROCEDURA**

FRAZA

Calculatorul informează că această entitate nu există deocamdată în BDR. Se cere consumămintul utilizatorului pentru a fi înregistrată. La răspuns negativ, întrebarea ENTITATE ? se repetă. Procedura se oprește prin răspunsul *". Utilizatorul răspunde că dorește înregistrarea acestei entități. Entitatea este înregistrată și poartă numărul 17 în BDR. Pentru această entitate se cer „cuvintele cheie” de regăsire. Se indexează după cuvintele „SOFTPLAN” și „UTILIZARE”. Se cere tipul entității înregistrate. Entitatea introdusă este o noțiune. Se cere data introducerii (sub forma aallzz). Se răspunde 15 octombrie 1981. În continuare utilizatorul face cunoșcut că dorește să înregistreze o frază care va fi legată de entitatea introdusă anterior. Calculatorul acceptă și comunică numărul pe care îl va purta această frază în BDR. Această frază leagă entitatea nou creată de o entitate existentă (presupunem, că entitatea <PROCEDURA> există dinainte în BDR). Predicatul [ESTE (CAZ PARTICULAR AL)] face ca toate propozițiile înregistrate la <PROCEDURA> să rămână valabile și în cazul noii entități cu precizările de rigoare. Se cere introducerea unei noi fraze. Faptul că nu s-a ivit nici un mesaj după formularea primei fraze înseamnă că toate entitățile menționate existau di-



Comenzi și răspunsuri la consola terminalului de teletransmisie:

Note:
explicative:

c : 32

u : PROCEDURA DE UTILIZARE
SOFTPLAN SE REFERA LA
SOFTPLAN

nainte în BDR; Altfel ar fi apărut mesajul NU EXISTA urmat de întrebarea SE INTRODUCÉ (1 = DA; 0 = NU) ca în cazul precedent.

Numărul pe care il va purta noua frază

Prin predicatul [SE REFERA LA] se face cunoscut că entitatea nou creată are legătură cu pachetul de programe SOFTPLAN a cărui descriere presupunem că s-a efectuat anterior prin fraze care se găsesc înregistrate la această entitate.

Ca urmare a conversației de mai sus, în BDR vor fi înregistrate următoarele fraze:

<PROCEDURA DE UTILIZARE SOFTPLAN>

FRAZA 31: <PROCEDURA DE UTILIZARE SOFTPLAN> [ESTE (CAZUL PARTICULAR AL)] <PROCEDURA>

FRAZA 32: <PROCEDURA DE UTILIZARE SOFTPLAN> [SE REFERA LA SOFTPLAN]
Imaginea structurii acestor fraze în BDR va arăta ca în figura 3.

Exemplul 2. În continuare, avind în vedere că <PROCEDURA DE UTILIZARE SOFTPLAN> este un caz particular al noțiunii mai generale numită <PROCEDURA>, să o utilizăm pe aceasta din urmă ca prototip pentru a introduce noi informații referitoare la noțiunea care ne interesează. Pentru aceasta este necesar să chemăm un nou program specializat care va dirija conversația cu calculatorul, program care poartă numele de „PARTICULARIZARE”.

u : EXEC PARTICULARIZARE ?

c : ENTITATE ?

u : PROCEDURA

c : DATA ?

u : 810901

c : FRAZA 25 : PROCEDURA

ARE SCOP

FRAZA 26 : PROCEDURA SE
COMPUNE DIN (SUCCESIUNE)

(PL) OPERATIE

FRAZA 27 : (FIECARE)

OPERATIE SE COMPUNE DIN
(ANSAMBLU) DEFINITIE (SI)

DURATA (SI) TERMEN (SI)

RESURSA (SI) EXECUTANT (SI)

RESPONSABIL (SI) CONDITIE/

Calculatorul întreabă numele entității și data textului din procedura generală ce urmează a fi particularizat.

Textul care ne interesează a fost introdus la 1 septembrie 1981. Urmează listarea acestui text.

RESTRICTIE (SI) PARAMETRU (SI)
REZULTAT POSIBIL (SI) COMANDA
FRAZA 28 : (FIECARE) REZULTAT
POSIBIL ESTE FUNCTIE DE
(UNII/UNELE) PARAMETRU (SI)
(UNII/UNELE) CONDITIE/RESTRICTIE
FRAZA 29 : (FIECARE) COMANDA
ESTE FUNCTIE DE (UN/O)
REZULTAT POSIBIL.

C; ATASAT ENTITATII ?

Calculatorul întreabă numele entității și data la care vor fi atașate frazele textului ce urmează a fi generat.

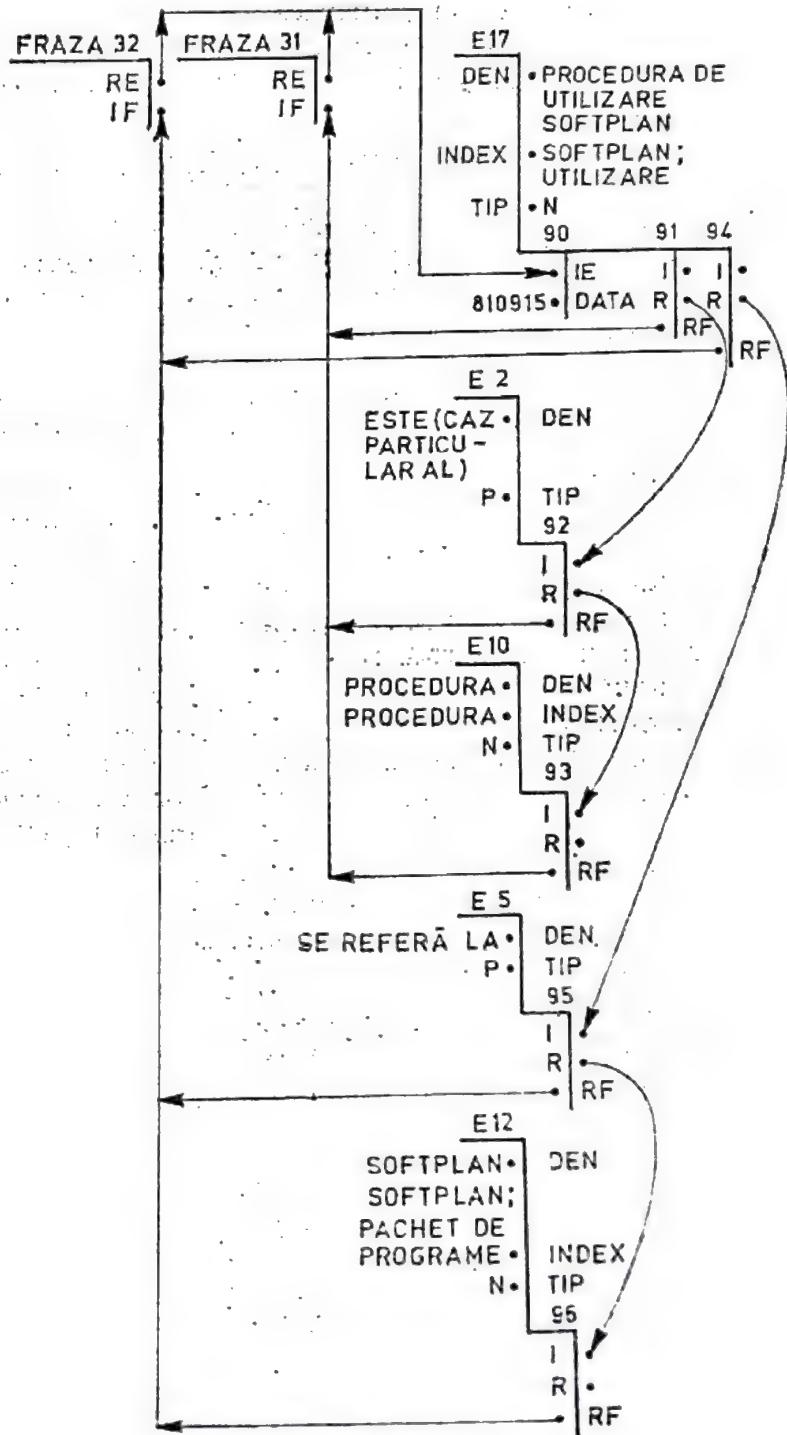


Fig. 3. — Imaginea structurii frazelor 31 și 32 create în urma dialogului din exemplul 1.

u : PROCEDURA DE UTILIZARE
SOFTPLAN •
c : DATA
u : 811015
c : FRAZA ?
u : 25

Se răspunde furnizind titlul și data
15 octombrie 1981.

c : SCOP ?
u : UTILIZARE (AL/A) SOFTPLAN
(IN) (PL) INTreprindere

În urma dialogului de mai sus se generează, prin urmare, următoarea frază :

〈PROCEDURA DE UTILIZARE SOFTPLAN〉

〈FRAZA 33〉 : 〈SCOP〉 [ESTE (IN ACEST CAZ)] 〈UTILIZARE〉 (AL/A) 〈SOFT-
PLAN〉 (IN) (PL) 〈INTreprindere〉

Structura frazelor generatoare (25) și generată (33) în BDR este ilustrată în figura 4.

Exemplul 3. Să continuăm operația incepută în exemplul precedent utilizând fraza 26 ca matriță generatoare. Vom avea următorul dialog :

c : FRAZA ?
u : 26

c : OPERATIE ?
u : COLECTAREA SI VERIFICAREA
DATELOR DE INTRARE
SOFTPLAN.
c : OPERATIE ?
u : INCARCAREA BAZEI DE DATE
SOFTPLAN

c : OPERATIE ?
u : VERIFICAREA LOGICA A BAZEI
DE DATE SOFTPLAN IN VEDEREA
OPTIMIZARII.
c : OPERATIE ?
u : PROCEDURA DE OPTIMIZARE
SOFTPLAN
c : OPERATIE ?
u : LISTAREA PROGRAMULUI DE
PRODUCTIE
c : OPERATIE ?
u : ADOPTAREA VARIANTEI
OPTIME
c : OPERATIE ?
u : *

Se cere numărul frazei generatoare. Se răspunde indicind fraza 26 (vezi exemplul 2). Fraza conține, aşa cum se poate constata, noțiunea 〈OPE-
RATIE〉 precedată de calificatorul (PL), acesta din urmă indicind (vezi tabela 2) mai multe obiecte de același fel care în cazul de față trebuie particularizate. Sirul de particularizări se termină cu comanda „*”.

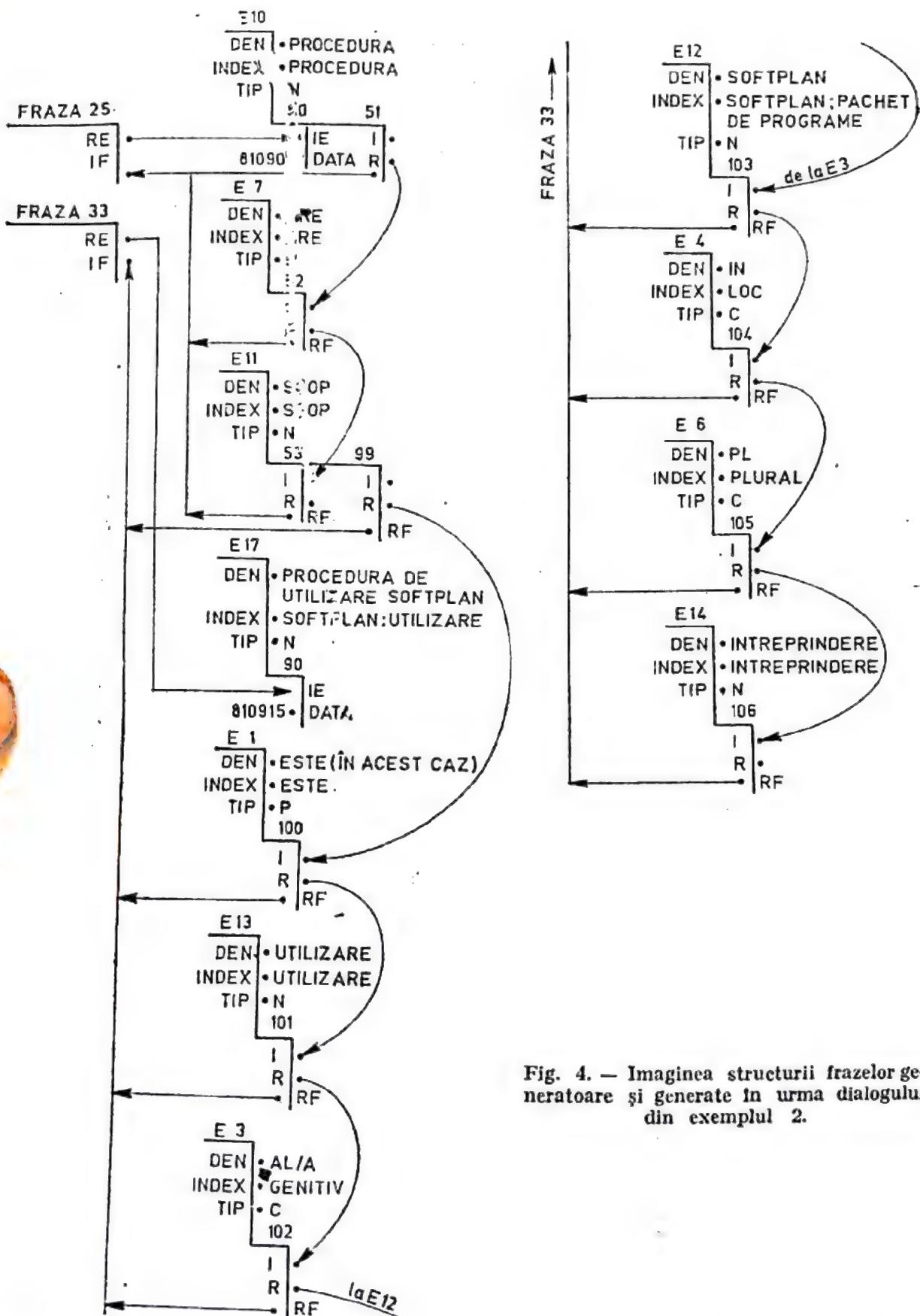


Fig. 4. — Imaginea structurii frazelor generatoare și generate în urma dialogului din exemplul 2.

Prin dialogul de mai sus se generează deci următorul text legat de entitatea care reprezintă totodată titlul textului :

⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩

⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩ [SE COMPUNE DIN (SUCCESIUNE)]

⟨COLECTAREA SI VERIFICAREA DATELOR DE INTRARE SOFTPLAN⟩ (URMAT DE) ⟨INCARCAREA BAZEI DE DATE SOFTPLAN⟩ (URMAT DE)

⟨VERIFICAREA LOGICA A BAZEI DE DATE SOFTPLAN IN VEDEREA OPTIMIZARII⟩ (URMAT DE) ⟨PROCEDURA DE OPTIMIZARE SOFTPLAN⟩ (URMAT DE) ⟨LISTAREA PROGRAMULUI DE PRODUCIE⟩ (URMAT DE) ⟨ADOPTAREA VARIANTEI OPTIME⟩

Exemplul 4. Să particularizăm, în continuare, fraza 27 (exemplul 2).

Se înregistrează următorul dialog :

c : FRAZA ?

u : 27

c : COLECTAREA SI VERIFICAREA

DATELOR DE INTRARE SOFTPLAN

SE COMPUNE DIN (ANSAMBLU)

DEFINITIE (1 = DA ; 0 = NU)

u : 1

DEFINITIE ESTE (CAZ PARTICULAR AL) COLECTAREA SI VERIFICAREA DATELOR

c : DURATA (1 = DA ; 0 = NU)

u : 1

c : TERMEN (1 = DA ; 0 = NU)

u : 0

c : RESURSA (1 = DA ; 0 = NU)

u : 1

c : EXECUTANT (1 = DA ; 0 = NU)

u : 0

c : INCARCAREA BAZEI DE DATE

SOFTPLAN SE COMPUNE DIN

(ANSAMBLU)

DEFINITIE (1 = DA ; 0 = NU)

u : 1

etc.

Rezultă următorul text legat de entitatea ⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩ :

⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩

⟨COLECTAREA SI VERIFICAREA DATELOR DE INTRARE SOFTPLAN⟩ [SE COMPUNE]

DIN (ANSAMBLU)] ⟨DEFINITIE⟩ (SI) ⟨DURATA⟩ (SI) ⟨RESURSA⟩ (SI) ...

⟨DEFINITIE⟩ [ESTE (CAZ PARTICULAR AL)] ⟨COLECTAREA SI VERIFICAREA DATELOR⟩

⟨INCARCAREA BAZEI DE DATE SOFTPLAN⟩ [SE COMPUNE DIN (ANSAMBLU)]

⟨DEFINITIE⟩ (SI) ...

etc.

Exemplul 5. Presupunem că entitatea **COLECTAREA SI VERIFICAREA DATELOR** a fost definită anterior prin următorul text :

COLECTAREA SI VERIFICAREA DATELOR

COLECTAREA SI VERIFICAREA DATELOR [ARE] (PL) **DATA** (SI) (PL) **FORMULAR**
FIECARE **DATA** [SE COMPUNE DIN (ANSAMBLU)] **DEFINITIE** (SI) **COD** (SI) **DE UNDE VINE SI CUM SE CALCULEAZA** (SI) **EXEMPLU DE CALCUL**
FIECARE **FORMULAR** [SE COMPUNE DIN (ANSAMBLU)] (UNII/UNELE) **DATA** (SI) **EXEMPLU DE COMPLETARE**.

Utilizând programul „PARTICULARIZARE” pentru entitatea **COLECTAREA SI VERIFICAREA DATELOR** se poate genera în continuare un text care particularizează și mai mult **PROCEDURA DE UTILIZARE SOFTPLAN**. Acest text, generat după același procedeu ca în exemplele 2, 3 și 4 anterioare, ar putea arăta astfel :

PROCEDURA DE UTILIZARE SOFTPLAN

COLECTAREA SI VERIFICAREA DATELOR DE INTRARE SOFTPLAN [ARE] (PL) **DATA** (SI) (PL) **FORMULAR**
DATA [SE COMPUNE DIN (ANSAMBLU)] **COD COMANDA** (SI) **COD PRODUS** (SI) **CANTITATE PRODUS** (SI) **TERMEN DE INCEPUT CEL MAI DEVREME** (SI) **TERMEN DE SFIRSIȚ CEL MAI TIRZIU** (SI) **COD POST DE LUCRU** (SI) **CAPACITATE POST PE PERIOADA** (SI) **INCARCARE PE PERIOADA** **FORMULAR** [SE COMPUNE DIN (ANSAMBLU)] **FORMULAR COMANDA** (SI) **FORMULAR 1 POST DE LUCRU** (SI) **FORMULAR 2 POST DE LUCRU** (SI) **FORMULAR 3 POST DE LUCRU** (SI) **FORMULAR 1 INCARCARE** (SI) **FORMULAR 2 INCARCARE** (SI) **FORMULAR 3 INCARCARE**
COD COMANDA [SE COMPUNE DIN (ANSAMBLU)] **DEFINITIE** (SI) **COD** (SI) **DE UNDE VINE SI CUM SE CALCULEAZA** (SI) **EXEMPLU DE CALCUL** **DEFINITIE** [ESTE (IN ACEST CAZ)] **COD** (AL/A) **COMANDA** **COD** [ESTE (IN ACEST CAZ)] **CODCOM**
DE UNDE VINE [ESTE (IN ACEST CAZ)] **SERVICIUL PLAN**
EXEMPLU DE CALCUL (AL/A) **COD COMANDA** [ESTE (IN ACEST CAZ)] **COM001**

FORMULAR COMANDA [SE COMPUNE DIN (ANSAMBLU)] **COD COMANDA** (SI) **COD PRODUS** (SI) **CANTITATE PRODUS** (SI) **TERMEN DE INCEPUT CEL MAI DEVREME** (SI) **TERMEN DE SFIRSIȚ CEL MAI TIRZIU** (SI) **EXEMPLU DE COMPLETARE**
EXEMPLU DE COMPLETARE (AL/A) **FORMULAR COMANDA** [ESTE (IN ACEST CAZ)] **COL 1-6 CODCOM ; COL 9-20 CODPROD ; COL 21-24 QT ; COL 25-28 '0002' ; COL 29-32 TID ; COL 33-36 TST**

Observații :

1° Calificatorul (UNII/UNELE) (vezi tabela 2) așezat înaintea entității **DATA** precizează că este vorba de alegerea unor din datele menționate pentru a alcătui subansamblul datelor care alcătuiesc un anume **FORMULAR**.

2º Predicatul [ARE] (vezi tabela 1) are un rol similar cu predicatul [SE COMPUNE DIN (ANSAMBLU)], toate entitățile care il urmează intrind în alcătuirea entității subiect.

Exemplul 6. Particularizarea noțiunilor, din aproape în aproape, poate fi continuată, în modul arătat, pînă la orice nivel. În cazul de față, presupunind că am dori să precizăm mai departe procedura de utilizare a pachetului de programe SOFTPLAN la Intreprinderea X, vom obține, plecînd de la textul generat în exemplul 5, următoarele fraze legate la entitatea <PROCEDURA DE UTILIZARE A SOFTPLAN LA INTREPRINDEREA X> care reprezintă, în acest caz, titlul noului text.

<PROCEDURA DE UTILIZARE SOFTPLAN LA INTREPRINDEREA X>

<PROCEDURA DE UTILIZARE SOFTPLAN LA INTREPRINDEREA X> [ESTE CAZ PARTICULAR AL] <PROCEDURA DE UTILIZARE SOFTPLAN>
<COD COMANDA> [SE COMPUNE DIN (ANSAMBLU)] <COM001> (SI) <COM002> (SI)

<FORMULAR COMANDA> [SE COMPUNE DIN (ANSAMBLU)] (DE LA) (1) (LA) (50)

- 1) <FORMULAR COMANDA> [ESTE (IN ACEST CAZ)]
<COL 1-6 'COM001'; COL 9-20 'PRODUS 014'; COL 21-24 '0050';
COL 25-28 '002'; COL 29-32 '0001'; COL 33-36 '0012'> (SI)
- (2) <FORMULAR COMANDA> [ESTE (IN ACEST CAZ)]
<COL 1-6 'COM002', COL 9-20 'PRODUS 015', COL 21-24 '0100';
COL 25-28 '0002'; COL 29-32 '0001'; COL 33-36 '0009'> (SI)

etc.

b. Utilizarea informațiilor înscrise în baza de date relațională

Să presupunem că, la un moment dat, există înscrise în baza de date relațională toate informațiile referitoare la *utilizarea pachetului de programe SOFTPLAN*, sub forma unor fraze legate de entități, în modul arătat în paragraful precedent (a). Ce utilizare putem da unor asemenea informații?

b.1. Utilizarea prin INDEX

Folosind caracteristica INDEX a entităților înscrise în baza de date putem cere calculatorului selectarea frazelor legate de entitățile care aparțin unui anumit domeniu care ne interesează. De exemplu, chemînd programul „SELECTINDEX” și cerînd selectarea după cuvintele cheie „COLECTAREA SI VERIFICAREA DATELOR” și „SOFTPLAN” și „INTREPRINDEREA X” calculatorul va căuta mai întii această combinație. În lipsa ei, va furniza informații de nivel mai general: „COLECTAREA SI VERIFICAREA DATELOR” și „SOFTPLAN” iar dacă și această combinație lipsește va lista toate frazele cuprinse în procedura generală de „COLECTAREA SI VERIFICAREA DATELOR”. Această căutare succesivă poate fi automatizată sau lăsată la latitudinea celui care beneficiază de serviciile calculatorului.

O altă formulare a condițiilor de selectare este legarea entităților prin funcția logică *sau*. Căutarea se face de la început pentru fiecare din cuvintele cheie menționate, după care se execută reuniunea ansamblelor selectate.

**b.2. Regăsirea informațiilor cu ajutorul predicatelor
[ESTE (CAZ PARTICULAR AL)] și [ESTE (IN ACEST CAZ)]**

Structura propusă mai sus pentru BDR permite regăsirea din aproape în aproape a unor informații din ce în ce mai detaliate cu privire la obiectul căutat. Modul de cercetare este exact invers ca în cazul înscrierii acestor informații folosind predicatul [ESTE (CAZ PARTICULAR AL)] și [ESTE (IN ACEST CAZ)] (vezi exemplele 2,3,4,5,6 din paragraful precedent).

Să presupunem, de pildă, că am dori să aflăm ce este cu entitatea <COD COMANDA> menționată în cadrul <PROCEDURA DE UTILIZARE SOFTPLAN LA INTreprinderea X> (vezi exemplul 6).

Legată de ultima entitate menționată mai sus, găsim fraza : <COD COMANDA> [SE COMPUNE DIN (ANSAMBLU)] <COM001> (SI) <COM002> (SI)...

Observăm apoi că :

<PROCEDURA DE UTILIZARE SOFTPLAN LA INTreprinderea X> [ESTE (CAZ PARTICULAR AL)] <PROCEDURA DE UTILIZARE SOFTPLAN>

Mergind la aceasta din urmă (exemplul 3) constatăm că :

<PROCEDURA DE UTILIZARE SOFTPLAN> [SE COMPUNE DIN (SUCESIUNE)]

<COLECTAREA ȘI VERIFICAREA DATELOR DE INTRARE SOFTPLAN> (URMAT DE)... etc.

iar aceasta [SE COMPUNE DIN (ANSAMBLU)] <DEFINITIE> (SI)... etc. (vezi exemplul 4).

Definiția colectării și verificării datelor de intrare SOFTPLAN arată că (vezi exemplul 4) <DEFINITIE> [ESTE (CAZ PARTICULAR AL)] <COLECTAREA ȘI VERIFICAREA DATELOR>.

Mergind la <COLECTAREA ȘI VERIFICAREA DATELOR> (exemplul 5) vedem că aceasta [ARE] (PL) <DATA> (SI) ... iar legat de <COLECTAREA ȘI VERIFICAREA DATELOR DE INTRARE SOFTPLAN> descoperim fraza

<DATA> [SE COMPUNE DIN (ANSAMBLU)] <COD COMANDA> (SI)... etc.

Așadar : <COD COMANDA> [FACE PARTE DIN] <DATA>, conform teoremei 2 (subcapitolul 3.3); <DATA> [FACE PARTE DIN] <COLECTAREA ȘI VERIFICAREA DATELOR>, conform corolarului teoremei 2 (subcapitolul 3.3); iar <COLECTAREA ȘI VERIFICAREA DATELOR> [ESTE (IN ACEST CAZ)] <COLECTAREA ȘI VERIFICAREA DATELOR DE INTRARE SOFTPLAN>, în baza transformării de echivalență permisă de teorema 1 (subcapitolul 3.3).

În același timp, <COLECTAREA ȘI VERIFICAREA DATELOR DE INTRARE SOFTPLAN> [FACE PARTE DIN] <PROCEDURA DE UTILIZARE SOFTPLAN>, conform teoremei 2 (subcapitolul 3.3), iar <PROCEDURA DE UTILIZARE SOFTPLAN> [ESTE (IN ACEST CAZ)] <PROCEDURA DE UTILIZARE SOFTPLAN LA INTreprinderea X>.

Astfel, legătura între **⟨COD COMANDA⟩** și entitatea **⟨PROCEDURA DE UTILIZARE SOFTPLAN LA INTreprinderEA X⟩** este clarificată. Se observă că, despre **⟨COD COMANDA⟩** se mai pot afla și alte informații. De pildă, tot din procedura generală de colectare și verificare a datelor (vezi exemplul 5) putem afla următoarele :

⟨COD COMANDA⟩ [SE COMPUNE DIN (ANSAMBLU)] **⟨DEFINIȚIE⟩** (SI) **⟨COD⟩** (SI) **⟨DE UNDE VINE⟩** (SI) **⟨CUM SE CALCULEAZĂ⟩** (SI) **⟨EXEMPLU DE CALCUL⟩** **⟨DEFINIȚIE⟩** [ESTE (IN ACEST CAZ)] **⟨COD⟩** (AL/A) **⟨COMANDA⟩** **⟨COD⟩** [ESTE (IN ACEST CAZ)] ...
etc.

b.3. Furnizarea unor informații pe baza prelucrării cadrelor relationale

Presupunem că ne interesează obținerea unor informații în legătură cu colectarea și verificarea datelor de intrare SOFTPLAN dar că, în BDR, nu există nici un fel de entitate **⟨COLECTAREA SI VERIFICAREA DATELOR DE INTRARE SOFTPLAN⟩**, **⟨COLECTAREA SI VERIFICAREA DATELOR⟩**, **⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩** sau **⟨PROCEDURA DE UTILIZARE SOFTPLAN LA INTreprinderEA X⟩**. Există însă, la entitatea **⟨SOFTPLAN⟩** o descriere a acestui pachet de programe prin intermediul unor fraze în limbajul INTELEC.

Cum procedăm în acest caz ?

Metoda adecvată este prelucrarea din aproape în aproape a cadrelor relationale.

Numim *cadrul relational* o frază în limbajul $L(G)$ definit anterior, în care pe post de subiect, predicat, complement sau calificatori ai acestora se află mai mult de o singură entitate, putind fi formate fraze din combinații ale acestora respectând poziția lor în cadrul frazei. Cadrelle relationale sunt deci funcții^{*)}, cu mențiunea că în loc de entități oarecare, variabilele pot lua un număr bine precizat de valori, specificate în cadrul frazei. Fraza care constituie cadrul relational poate fi, la limită, lipsită de predicat, cu alte cuvinte, poate avea forma unei entități (cu calificatori) sau siruri de entități.

Să revenim acum la problema noastră și să formulăm un cadrul relational pentru subiectul care ne interesează :

$$\left\{ \begin{array}{l} \langle \text{COLECTARE} \rangle ; \\ \langle \text{VERIFICARE} \rangle \end{array} \right\} \text{(AL/A) (PL)} \langle \text{DATA (INFORMATIE)} \rangle \left\{ \begin{array}{l} \langle \text{DE} \rangle ; \\ \langle \text{PENTRU} \rangle ; \\ \langle \text{IN} \rangle \end{array} \right\} \langle \text{INTRARE} \rangle \quad (3.31)$$

Pentru prelucrarea acestui cadrul vom proceda în felul următor :

Faza I

Se descoperă predicatele legate de entități prin predicatul intern **[ESTE DERIVAT DIN]*** (vezi tabela 1).

$$\begin{aligned} \langle \text{COLECTARE} \rangle & [ESTE DERIVAT DIN] * \langle \text{COLECTEAZA} \rangle & (3.32) \\ \langle \text{VERIFICARE} \rangle & [ESTE DERIVAT DIN] * \langle \text{VERIFICA} \rangle. \end{aligned}$$

^{*)} Vezi subcapitolul 3.3.

Cu aceste predicate se formează fraza :

$\langle \text{SE} \rangle \left\{ \begin{array}{l} [\text{COLECTEAZA}] ; \\ [\text{VERIFICA}] \end{array} \right\} \text{(PL)} \langle \text{DATA (INFORMATIE)} \rangle \left\{ \begin{array}{l} (\text{DE}) ; \\ (\text{PENTRU}) ; \\ (\text{IN}) \end{array} \right\} \langle \text{IN-} \right. \\ \left. \text{TRA-} \right. \\ \left. \text{RE} \rangle. \quad (3.33)$

Faza a II-a

Se identifică entitatea-complement, în cazul nostru

$\langle \text{DATA (INFORMATIE)} \rangle \left\{ \begin{array}{l} (\text{DE}) ; \\ (\text{PENTRU}) ; \\ (\text{IN}) \end{array} \right\} \langle \text{INTRARE} \rangle \quad (3.34)$

Pentru entitatea-cadru de mai sus se caută în **<SOFTPLAN>** toate frazele care o conțin împreună cu unul din predicatele :

[SE COMPUNE DIN (ANSAMBLU)], [FACE PARTE DIN], [ESTE (CAZ PARTICULAR AL)]
[ESTE (IN ACEST CAZ)]. Presupunem că se găsește fraza :

$\langle \text{DATA (INFORMATIE)} \rangle \text{ (DE)} \langle \text{INTRARE} \rangle [\text{SE COMPUNE DIN (ANSAMBLU)}] \quad (3.35)$
 $\langle \text{DATE COMENZI} \rangle \text{ (SI)} \langle \text{DATE POSTURI} \rangle \text{ (SI)} \langle \text{DATE INCARCARI} \rangle$

O a doua prelucrare a complementului, de astă dată pentru fraza de mai sus, duce la :

$\langle \text{DATE COMENZI} \rangle [\text{SE COMPUNE DIN (ANSAMBLU)}] \langle \text{COD COMANDA} \rangle \text{ (SI)} \quad (3.36)$
 $\langle \text{COD PRODUS} \rangle \text{ (SI)} \langle \text{CANTITATE PRODUS} \rangle \text{ (SI)} \langle \text{TERMEN DE INCEPUT CEL MAI DEVREME} \rangle \text{ (SI)} \langle \text{TERMEN DE SFIRSIIT CEL MAI TIRZIU} \rangle$
 $\langle \text{DATE POSTURI} \rangle [\text{SE COMPUNE DIN (ANSAMBLU)}] \langle \text{COD POST DE LUCRU} \rangle \text{ (SI)}$
 $\langle \text{CAPACITATE PE PERIOADA} \rangle$

$\langle \text{DATE INCARCARI} \rangle [\text{SE COMPUNE DIN (ANSAMBLU)}] \langle \text{COD COMANDA} \rangle \text{ (SI)} \langle \text{COD PRODUS} \rangle \text{ (SI)} \langle \text{COD POST DE LUCRU} \rangle \text{ (SI)} \langle \text{INCARCARE PE PERIOADA} \rangle.$

O a treia prelucrare a complementului, pentru frazele de mai sus, ar duce la descoperirea definițiilor noțiunilor complement, a codurilor acestora, de unde vine și cum se calculează fiecare noțiune a exemplelor de calcul etc.

Procesul poate continua în acest fel pînă la nivelul de detaliere dorit.

Faza a III-a

Se identifică toate frazele care utilizează unul din predicatele [COLECTEAZA] sau [VERIFICA], avînd drept complement cadrul relațional (3.34) de mai sus sau oricare din noțiunile complement descoperite în faza a II-a (3.35), (3.36), de exemplu :

$[\langle \text{SE} \rangle \langle \text{COLECTEAZA} \rangle \langle \text{DATE COMENZI} \rangle \text{ (PRIN)} \langle \text{INSCRIERE} \rangle \text{ (PE)} \quad (3.37)$

$\langle \text{FORMULAR COMANDA} \rangle$

$\langle \text{SE} \rangle \langle \text{VERIFICA} \rangle \langle \text{CAPACITATE PE PERIOADA} \rangle \text{ (CU)} \langle \text{PLAN REPARATII UTILAJE} \rangle$
etc.

O prelucrare ulterioară a complementului la frazele de mai sus treceind în prealabil prin faza de derivare, va duce la explicitarea noțiunilor descoperite, de exemplu :

$\langle \text{INSCRIERE} \rangle [ESTE DERIVAT DIN]^* \langle \text{INSCRIE} \rangle$ (3.38)

Cadrul relațional $\langle \text{INSCRIERE} \rangle$ (PE) $\langle \text{FORMULAR COMANDA} \rangle$ din (3.37) se transformă în fraza :

$\langle \text{SE} \rangle [\text{INSCRIE}] (\text{PE}) \langle \text{FORMULAR COMANDA} \rangle$ (3.39)

în care $\langle \text{FORMULAR COMANDA} \rangle$ este explicitat mai departe prin :

$\langle \text{FORMULAR COMANDA} \rangle [\text{SE COMPUNE DIN (ANSAMBLU)}] \langle \text{COD COMANDA} \rangle$

(SI) $\langle \text{COD PRODUS} \rangle$ (SI) $\langle \text{CANTITATE PRODUS} \rangle$ (SI) $\langle \text{PRIORITATE} \rangle$

(SI) $\langle \text{TIIMP DE INCEPUT CEL MAI DEVREME} \rangle$ (SI) $\langle \text{TIIMP DE SFIRSI} \rangle$
CEL MAI TIRZIU (3.40)

O prelucrare a datelor de tipul aceleia descrise mai sus ar putea avea la bază următorul algoritm *)

$I \leftarrow 1$

repetă pînă cînd $I > N$

pentru toate || cadrele relaționale din 'LISTA CADRELOR' ||

pentru toate || frazele care conțin cadrul curent împreună cu unul din predicatele care figurează în 'LISTA PREDICATELOR' ||

|| Listeață frazele găsite ||

|| Completează 'LISTA FRAZELOR' cu frazele găsite ||

□

□

dacă $I = N$ atunci stop

pentru toate || frazele din 'LISTA FRAZELOR' ||

|| Separă cadrele relaționale complement sau subiect ** ||

|| Completează 'LISTA CADRELOR' ||

|| Șterge fraza curentă din 'LISTA FRAZELOR' ||

□

pentru toate || Cadrele relaționale din 'LISTA CADRELOR' ||

|| Prelucră cadrul curent găsind predicatele derive ||

|| Include predicatele derive în 'LISTA PREDICATELOR' ||

|| Înlocuiește cadrul curent cu cel rămas în urma extragerii entității derive ||

□

$I \leftarrow I + 1$

□

*) Pentru reprezentarea tuturor algoritmilor care figurează în această lucrare s-au folosit notațiile din tabela 3.

**) Separarea se face în funcție de predicat.

De exemplu dacă este vorba de predicatul [SE COMPUNE DIN (ANSAMBLU)], se includ în 'LISTA CADRELOR', una cite una, noțiunile complement care urmează. Dacă predicatul frazei este [ESTE (IN ACEST CAZ)], se include în 'LISTA CADRELOR' complementul acestei fraze.

Dacă predicatul este [FACE PARTE DIN] sau [ESTE (CAZ PARTICULAR AL)] se include în 'LISTA CADRELOR' subiectul.

Dacă predicatul este derivat dintr-o noțiune se include în 'LISTA CADRELOR' complementul. etc.

Tabelă 3

NOTAȚII UTILIZATE PENTRU REPREZENTAREA ALGORITMILOR
DE PRELUCRARE AUTOMATĂ A DATELOR

Nr. crt.	Simbol	Semnificație	Exemplu
1	\leftarrow	Alocare a valorii din dreapta, variabilei aflată în stînga acestui semn.	$I \leftarrow 1$
2	\square	Semn de terminare a unei fraze.	$dacă I > N$ $atunci \dots \square$
3	$\ \dots \dots \ $	Textul încadrat de aceste semne reprezintă explicația succintă în limbaj natural, sau titlul unui subalgoritm care apare în algoritmul principal.	$\ $ Frazele din 'LISTA FRAZELOR' $\ $
4	$::=$	Explicitare a unei operații sau algoritm, avînd simbolul sau titlul plasat în stînga iar expansiunea (șir de operații sau text) plasate în dreapta acestui semn.	$\ $ Condiție $\ ::=$ $I > N$
5	<i>repetă pînă cînd</i> $\ $ Condiție $\ $ $\ $ Corp $\ $ \square	Operațiile înschise în $\ $ Corp $\ $ se repetă de atîtea ori pînă cînd se indeplinește $\ $ condiție $\ $. Se trece apoi la operația următoare din algoritm.	$I \leftarrow 1$ <i>repetă pînă</i> <i>cînd</i> $I > N$ $i \leftarrow i + 1$ \square
6	<i>pentru toate</i> $\ $ Entitate $\ $ $\ $ Corp $\ $ \square	Se face accesul pe rînd la toate entitățile definite prin $\ $ Entitate $\ $. Pentru fiecare din ele se execută operațiile explicitate prin $\ $ Corp $\ $. După epuizarea tuturor entităților se trece la operația următoare din algoritm.	<i>pentru toate</i> $\ $ Frazele din 'LISTA FRAZELOR' $\ $ $\dots \dots$ \square

Tabela 3 (continuare)

Nr. crt.	Simbol	Semnificație	Exemplu
7	<i>dacă</i> $\ $ Condiție $\ $	Dacă condiția exprimată prin atunci $\ $ Comandă ₁ $\ $ $\ $ Condiție $\ $ este indeplinită (altfel $\ $ Comandă ₂ $\ $) atunci se execută $\ $ Comandă ₁ $\ $ dacă nu, atunci se execută $\ $ Comandă ₂ $\ $. (Dacă altfel $\ $ Comandă ₂ $\ $ lipsește, la neindeclinarea condiției se trece la operația următoare din algoritm)	<i>dacă I=N</i> atunci <i>stop</i>
8	<i>mergi la</i>	Comanda de salt necondiționat la executarea operației specificată în continuare.	<i>mergi la</i> <i>Operația n</i>
9	<i>execută</i>	Se execută comenziile care se află inscrise la entitatea care urmează, după care se revine la algoritmul principal.	<i>execută</i> \langle SUBPROGRAM \rangle
10	<i>întoarce</i>	Comandă de întoarcere la algoritmul principal.	
11	<i>stop</i>	Terminarea executării algoritmului.	

Pentru a utiliza algoritmul de mai sus trebuie fixată în prealabil valoarea lui N care, aici are semnificația numărului de niveluri pentru care se întinde explorarea și ne permite să lărgim sau să limităm, după voie, sfera noțiunilor explorate. Se înțelege că dacă baza de date este suficient de bogată, explorarea poate să „explodeze” după un număr de cicluri ajungîndu-se la un volum foarte mare de fraze care au din ce în ce mai puțină tangență cu cadrul inițial.

Tot în vederea utilizării acestui algoritm trebuie să inițializăm 'LISTA PREDICATELOR' (Se vede că dacă această listă este vidă algoritmul este ineficient). Semnificația acestei liste este următoarea : dacă, în exemplul de mai sus am introduce, inițial, să spunem, numai predicatul [SE COMPUNE DIN (ANSAMBLU)], am limita de la început frazele căutate numai la această formă. Este evident că, în acest caz, o frază de tipul : \langle COD COMANDA \rangle [FACE PARTE DIN] \langle DATE COMENZI \rangle ar fi ignorată deși semnificația ei ar fi identică (vezi teorema 2, subcapitolul 3.3) cu a frazei \langle DATE COMENZI \rangle [SE COMPUNE DIN (ANSAMBLU)] \langle COD COMANDA \rangle (S1)... care ar fi selectată.

O soluție ar fi, desigur includerea de la sine și a predicatului [FACE PARTE DIN] în 'LISTA PREDICATELOR' totuși, trebuie remarcat faptul că includerea *tuturor* predicatelor (din tabela 1) în această listă este mult prea costisitoare din punct de vedere al timpului de explorare în baza de date și, de ce să nu recunoaștem, de multe ori inutilă.

b.4. *Stabilirea unor legături logice între două informații oarecare din BDR*

Să presupunem următorul text inseris în BDR

⟨TEXT B4⟩

F1 : ⟨CONDUCEREA OPERATIVA A PRODUCTIEI⟩ [ESTE (CAZ PARTICULAR AL)]
 ⟨PROCES DE CONDUCERE⟩

F2 : ⟨PROCES DE CONDUCERE⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨PREGATIREA FABRICATIEI⟩ (SI) ⟨DETERMINAREA TIPULUI DE PRODUCTIE SI A FORMELOR DE ORGANIZARE⟩ (SI) ⟨CALCULUL PARAMETRILOR⟩ (SI) ⟨ORDONANTAREA FABRICATIEI⟩

F3 : ⟨CONDUCEREA OPERATIVA A PRODUCTIEI [ESTE (CAZ PARTICULAR AL)]
 ⟨CONDUCERE OPERATIVA⟩

F4 : ⟨CONDUCERE OPERATIVA⟩ [ESTE (CAZ PARTICULAR AL)] ⟨CONDUCEREA INTreprinderii⟩

F5 : ⟨PROCES DE CONDUCERE⟩ [ESTE (CAZ PARTICULAR AL)] ⟨PROCES⟩

F6 : ⟨PROCES DE CONDUCERE⟩ [SE REFERA LA] ⟨CONDUCEREA INTreprinderii⟩

Se poate pune următoarea întrebare: ce legătură există în trei: ⟨CONDUCEREA OPERATIVA A PRODUCTIEI⟩ și 2: ⟨CONDUCEREA INTreprinderii⟩?

Pentru a răspunde, se poate proceda în felul următor:

1° Se caută mai întii frazele în care apare entitatea ⟨CONDUCEREA OPERATIVA A PRODUCTIEI⟩. Se găsește F1 și F3. Notăm acest proces cu $1 \rightarrow (F_1, F_3)$.

2° Se caută apoi frazele care utilizează ⟨CONDUCEREA INTreprinderii⟩. Se găsesc F4 și F6, prin urmare $2 \rightarrow (F_4, F_6)$.

Din compararea celor două subansamblă $S_1 = \{F_1, F_3\}$ și $S_2 = \{F_4, F_6\}$ nu rezultă nici o frază care ar utiliza simultan ambele noțiuni cu alte cuvinte, intersecția lor este vidă: $S_1 \cap S_2 = \emptyset$.

Concluzia. Nu există nici o legătură *directă* între noțiunile ce ne interesează.

Să procedăm acum la identificarea legăturilor *de ordinul 1* (cu un element intermediar).

3° Se ia fiecare element din primul subansamblu (apărținând frazelor F_1 și F_3) și se listează toate frazele care utilizează aceste noțiuni. În cazul nostru, entitățile (diferite de cele inițiale) sunt 3: ⟨PROCES DE CONDUCERE⟩ și 4: ⟨CONDUCEREA OPERATIVA⟩. Notăm $F_1 \rightarrow 3$, $F_3 \rightarrow 4$. Referitor la prima entitate găsim frazele F_2 , F_5 și F_6 care o utilizează iar referitor la cea de-a doua frazele F_3 și F_4 în care intră entitatea 4. Rezultă că $3 \rightarrow (F_2, F_5, F_6)$ iar $4 \rightarrow (F_3, F_4)$.

Intersectând pe rînd subansambllele $S_3 = \{F_2, F_5, F_6\}$ și $S_4 = \{F_3, F_4\}$ cu $S_2 = \{F_4, F_6\}$ obținem două intersecții nevide și anume:

$$S_3 \cap S_2 = \{F_6\}, \text{ iar } [S_4 \cap S_2 = \{F_4\}].$$

Prin urmare am stabilit două legături de ordinul 1 între noțiunile 1 și 2, cu ajutorul noțiunilor intermediare 3, împreună cu frazele F_1 și F_6 .

pentru prima legătură și, respectiv 4, împreună cu frazele F_3 și F_4 pentru cea de-a doua, ceea ce se poate exprima simbolic prin :

$$1 \rightarrow F_1 \rightarrow 3 \rightarrow F_6 \rightarrow 2, \text{ și} \quad (3.41)$$

$$1 \rightarrow F_3 \rightarrow 4 \rightarrow F_4 \rightarrow 2 \quad (3.42)$$

4° Continuînd explorarea frazelor descoperite în pasul precedent și dînd la iveală alte noțiuni, putem stabili – dacă printre acestea din urmă figurează și cea de-a doua entitate – legături de *ordinul 2* între entitățile 1 și 2 inițiale. Procesul poate continua astfel pînă la epuizarea stocului de fraze existent în baza de date. În cazul de față continuarea nu mai are sens deoarece stocul de fraze disponibile a fost în întregime parcurs cu ocazia stabilirii legăturilor de ordinul 1.

b.5. Generalizarea unor cazuri particulare și crearea unor informații sintetice

Să ne referim la entitatea **CONDUCEREA OPERATIVA A PRODUCȚIEI LA INTreprinderea X** și să presupunem că de această entitate sunt legate următoarele fraze care arată cum a decurs acest proces într-o anumită perioadă la această întreprindere :

CONDUCEREA OPERATIVA A PRODUCȚIEI LA INTreprinderea X

⟨SE⟩ [EFFECTUEAZA] ⟨PREGATIREA FABRICATIEI⟩ (LA DATA) ...
⟨PREGATIREA FABRICATIEI⟩ [ARE] ⟨DURATA⟩ (DE) ...
⟨SE⟩ [EFFECTUEAZA] ⟨PREGATIREA FABRICATIEI⟩ (DE CATRE) ⟨PERSOANA 1⟩ (SI) ⟨PERSOANA 2⟩ (SI) ...
⟨PREGATIREA FABRICATIEI⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨PREGATIREA TEHNOLOGIILOR DE FABRICATIE⟩ (AL/A) (PL) ⟨PRODUS⟩ (SI) ⟨ALCATUIREA LISTEI DE PRODUSE PENTRU ANUL DE PLAN⟩ (SI) ⟨ALCATUIREA DOCUMENTATIEI DE LANSARE⟩ (SI) ...

⟨SE⟩ [EFFECTUEAZA] ⟨DETERMINAREA TIPULUI DE PROducțIE ȘI A FORMELOR DE ORGANIZARE⟩ (PENTRU) ⟨PRODUS 1⟩ (SI) ⟨PRODUS 2⟩ (SI) ...
⟨SE⟩ [EFFECTUEAZA] ⟨DETERMINAREA TIPULUI DE PROducțIE ȘI A FORMELOR DE ORGANIZARE⟩ (LA DATA) ...
⟨DETERMINAREA TIPULUI DE PROducțIE ȘI A FORMELOR DE ORGANIZARE⟩ [ARE] ⟨DURATA⟩ (DE) ...
⟨SE⟩ [EFFECTUEAZA] ⟨DETERMINAREA TIPULUI DE PROducțIE ȘI A FORMELOR DE ORGANIZARE⟩ (DE CATRE) ⟨PERSOANA 11⟩ (SI) ⟨PERSOANA 12⟩ (SI) ...
⟨DETERMINAREA TIPULUI DE PROducțIE ȘI A FORMELOR DE ORGANIZARE⟩ [SE COMPUNE DIN (ANSAMBLU)] ...

⟨SE⟩ [EFFECTUEAZA] ⟨CALCULUL PARAMETRILOR⟩ (PENTRU) ⟨PRODUS 1⟩ (SI) ⟨PRODUS 2⟩ (SI) ...
⟨SE⟩ [EFFECTUEAZA] ⟨CALCULUL PARAMETRILOR⟩ (LA DATA) ...
⟨CALCULUL PARAMETRILOR⟩ [ARE] ⟨DURATA⟩ (DE) ...



⟨SE⟩ [EFFECTUEAZA] ⟨CALCULUL PARAMETRILOR⟩ (DE CATRE)
 ⟨PERSOANA 21⟩ (SI) ⟨PERSOANA 22⟩ (SI) ...
 ⟨CALCULUL PARAMETRILOR⟩ [SE COMPUNE DIN (ANSAMBLU)] ...

⟨SE⟩ [EFFECTUEAZA] ⟨ORDONANTAREA FABRICATIEI⟩ (PENTRU PERIOADA) ...
 ⟨SE⟩ [EFFECTUEAZA] ⟨ORDONANTAREA FABRICATIEI⟩ (PENTRU) ⟨PRODUS 1⟩
 (SI) ⟨PRODUS 2⟩ (SI) ...

⟨ORDONANTAREA FABRICATIEI⟩ [ARE] ⟨DURATA⟩ (DE) ...

⟨SE⟩ [EFFECTUEAZA] ⟨ORDONANTAREA FABRICATIEI⟩ (DE CATRE)
 ⟨PERSOANA 31⟩ (SI) ⟨PERSOANA 32⟩ (SI) ...

⟨SE⟩ [EFFECTUEAZA] ⟨ORDONANTAREA FABRICATIEI⟩ (IN CONFORMITATE CU)
 ⟨PROCEDURA DE UTILIZARE SOFPLAN⟩.

Dacă deserierea de mai sus se întâlnește pentru mai multe întreprinderi, fie ele ⟨INTREPRINDEREA X⟩, ⟨INTREPRINDEREA Y⟩, ⟨INTREPRINDEREA Z⟩ etc. se poate face o generalizare a noțiunii de ⟨CONDUCERE OPERATIVA A PRODUCTIEI⟩ în felul următor :

1º Se cercetează care anume entități ⟨x⟩ se încadrează în schema unor fraze de tipul :

⟨SE⟩ [EFFECTUEAZA] ⟨x⟩ { (LA DATA)
 (DE CATRE)
 (PENTRU PERIOADA)
 etc. } ..., (3.43)

sau, de tipul

⟨x⟩ [ARE] { ⟨DURATA⟩ (DE)
 ⟨NEVOIE⟩ (DE)
 etc. } ..., (3.44)

sau, de tipul :

⟨x⟩ [SE COMPUNE DIN (ANSAMBLU)] ..., (3.45)

cu condiția ca ⟨x⟩ să nu rezulte din detalierea unor entități, cu alte cuvinte să nu fi fost obiectul unor fraze de tipul :

⟨a⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨x⟩ (SI) ... (3.46)

sau

⟨x⟩ [FACE PARTE DIN] ⟨a⟩ (3.47)

în care ⟨a⟩ este o entitate oarecare de un grad de generalitate mai mare decit ⟨x⟩.

Pentru exemplul nostru se găsesc entitățile :

⟨PREGATIREA FABRICATIEI⟩, ⟨DETERMINAREA TIPULUI DE PRODUCTIE SI A FORMELOR DE ORGANIZARE⟩, ⟨CALCULUL PARAMETRILOR⟩ și ⟨ORDONANTAREA FABRICATIEI⟩.

2º Pentru entitățile care satisfac condițiile de la punctul 1º se poate alcătui următoarea frază :

⟨CONDUCEREA OPERATIVA A PRODUCTIEI⟩ { [SE COMPUNE DIN (ANSAMBLU)] ;
 [ARE] } ; (3.48)

〈PREGATIREA FABRICATIEI〉 (SI) 〈DETERMINAREA TIPULUI DE PRODUC-
TIE SI A FORMELOR DE ORGANIZARE〉 (SI) 〈CALCULUL PARAMETRILOR〉
 (SI) 〈ORDONANTAREA FABRICATIEI〉

care reprezintă, aproximativ fraza *F2* din exemplul b.4, de mai sus, și care are totodată un grad de generalitate superior acelora din textul inițial.

3º Procedind în același fel se poate trece la generalizarea unor aspecte ale părților componente ale entității inițiale, cu alte cuvinte se pot trage unele concluzii referitoare la :

- succesiunea operațiilor (prin compararea *datelor* la care are loc fiecare din operații) ;
- durata (medie) a fiecărui tip de operație (prin efectuarea *mediei duratelor* aceleiași operații la diferite întreprinderi) ;
- persoanele care execută fiecare tip de operație (*numărul* lor precum și *calificarea* lor dacă aceasta din urmă este specificată pentru diverse întreprinderi) ;
- părțile componente (fazele) fiecărui tip de operație în parte, prin luarea în considerare a frazelor de forma 〈*tip de operație 1*〉 [SE COMPUNE DIN (ANSAMBLU)] 〈*faza 11*〉 (SI) 〈*faza 12*〉 (SI) . . . , de exemplu (vezi textul inițial) :

〈PREGATIREA FABRICATIEI〉 [SE COMPUNE DIN (ANSAMBLU)] 〈PREGATIREA
TEHNOLOGILOR DE FABRICATIE〉 (SI) 〈ALCATUIREA LISTEI DE PRO-
DUSE PENTRU ANUL DE PLAN〉 (SI) 〈ALCATUIREA DOCUMENTATIEI DE
LANSARE〉 (SI) . . .

CAPITOLUL

4

Planificarea comportamentului în cadrul modelului INTELEC

Planificarea presupune, de regulă, o acțiune de anticipare specifică mișcii omenesti. Nu putem planifica comportamentul decidentului în rezolvarea unei probleme dacă nu prevedem dinainte toate situațiile posibile indicind cea mai bună cale de urmat în fiecare din acestea.

Și totuși omul „face față” și unor cazuri neprevăzute. Ba, am putea spune că în viața de toate zilele cea mai mare parte a acțiunilor unui individ nu sunt planificate. Decizia se adaptă pe loc și este mai bună sau mai puțin bună după cum puterea de judecată, memoria, experiența, universul gândirii individuale sunt mai adaptate sau mai puțin adaptate rezolvării unor asemenea probleme. Dacă ne referim însă la activitatea economică a întreprinderii, putem spune că aceasta se desfășoară într-un cadru mult mai organizat. Se cunosc dinainte scopul, obiectivele, modalitățile de acțiune, structura și funcționarea normală a compartimentelor, marea majoritate a abaterilor posibile. Universul de acțiune este, în acest caz, mult mai limitat. Întrebarea care se pune atunci, în mod firesc, este: nu cumva am putea reprezenta printr-un mijloc oarecare, acest univers într-un calculator, contribuind apoi, cu competență, la soluționarea problemelor de conducere economică a întreprinderii? În cele ce urmează am încercat să răspundem afirmativ acestor întrebări, aplicând modelul INTELEC unei arii mai restrinse de activitate din cadrul întreprinderii și anume în conducerea operativă a producției cu ajutorul pachetelor de programe create special în acest scop. S-au creat o procedură de lucru, un **PROGRAM PRINCIPAL** și un **SUBPROGRAM** a căror aplicare pe baza unui algoritm general poate sta la baza tuturor deciziilor legate de utilizarea pachetelor de programe pentru producție. Ca exemplu a fost luat pachetul de programe SOFTPLAN a cărui descriere detaliată este dată în cadrul capitolului 6.

4.1. Generalități

Vom numi *planificare a comportamentului* orice acțiune imaginabilă menită să conduce la soluționarea unei probleme ([67] p. 7). În sensul teoriei deciziei (vezi capitolul 2) se poate preciza că planificarea constă din alcătuirea unui *plan de acțiune* care vizează un anumit *scop* și care poate conduce sau nu la o *situatie dinainte prevăzută*. Dacă rezultatul se inseră într-una din aceste situații, urmează, imediat, o nouă acțiune previzibilă și.a.m.d., pînă la atingerea țelului final. Ce se întimplă însă dacă rezultatul nu face parte din mulțimea cunoscută pentru care există o comandă stabilită anterior? Aici intervine *inteligenta* — naturală sau artificială — care conduce la o *decizie inedită*.

Întrebarea care urmează în mod firesc este următoarea: poate un automat programabil (în cazul de față calculatorul electronic) să preia această sarcină și să dea naștere la o decizie eficientă? Desigur, un răspuns absolut ar fi prematur, mai ales dacă ne gîndim la faptul că unele încercări în acest sens s-au soldat cu eșecuri mai mult sau mai puțin descurajatoare ([67] p. 7–14). Îndrăznim să afirmăm totuși, că în cazul unor domenii suficient de restrînse — cum ar fi acela al conducerii operative a producției — și cu condiția de a respecta în orice acțiune o procedură generală cunoscută* ([13] p. 128–130), [64], calculatorul poate furniza decizii bune sau poate ajuta omul în luarea acestora. Pentru a susține afirmația de mai sus ne vom servi de modelul INTELEC al cărui limbaj a fost definit în cadrul capitolului anterior.

4.2. Planificarea comportamentului în cazul utilizării optime a unui pachet de programe pentru conducerea operativă a producției

Fie problema utilizării optime a unui pachet de programe, în general, în cadrul procesului de conducere operativă a producției.

Presupunem că în baza de date relațională (BDR) există, inscrisă sub formă de text, o procedură generală de acțiune valabilă pentru orice proces de decizie de acest tip, procedură care arată astfel:

〈PROCEDURA〉

〈PROCEDURA〉 [ARE] 〈SCOP〉
 〈SCOP〉 [SE COMPUNE DIN (ANSAMBLU)] (PL) 〈OBIECTIV〉

* Referitor la modalitatea de a rezolva o problemă recurgînd la o procedură generală sau combinînd o procedură generală cu alta specifică, menționăm că aceasta pare a fi o modalitate proprie sistemului nervos al omului (vezi [13]). Numai că procedurile generale și specifice precum și felul în care sunt ele combinate de creierul omului nu au fost încă descoperite. Încercările în acest sens ale lui Newell și Simon au dus la construirea „rezolvatorului de probleme generale” (General Problem Solver) (vezi [64]) care însă s-a dovedit incapabil să rezolve o clasă largă de probleme, avînd de fapt o utilizare mult mai restrînsă decît se credea la început.

(FIECARE) <OBIECTIV> [ESTE (IN ACEST CAZ)] (UN/O) <REZULTAT POSIBIL>
 <PROCEDURA> [SE COMPUNE DIN (ANSAMBLU)] (PL) <OPERATIE>
 <PRIMA OPERATIE> [ESTE (IN ACEST CAZ)] (UN/O) <OPERATIE>
 <OPERATIA URMATOARE> ESTE (IN ACEST CAZ] (UN/O) <OPERATIE>
 (FIECARE) <OPERATIE> [SE COMPUNE DIN (ANSAMBLU)] <DEFINITIE> (SI)
 <DURATA> (SI) <TERMEN> (SI) <EXECUTANT> (SI) <RESPONSABIL> (SI)
 <CONDITIE/RESTRICTIE>¹⁾ (SI) <PARAMETRU>²⁾ (SI) <REZULTAT POSIBIL>³⁾
 (SI) <SELECTIA OPERATIEI URMATOARE>⁴⁾ (SI) <FIXARE PARAMETEI>
 (SI) <FIXARE CONDITII/RESTRICTII>
 <REZULTAT POSIBIL> [ESTE FUNCTIE DE]⁵⁾ (PL) <PARAMETRU> (SI) (PL) <CONDITIE/RESTRICTIE>
 <SELECTIA OPERATIEI URMATOARE> [ESTE FUNCTIE DE] (PL) <REZULTAT POSIBIL> (SI) (PL) <REZULTAT OBTINUT> (SI) (PL) <PARAMETRU> (SI) (PL) <CONDITIE/RESTRICTIE>

în care :

¹⁾ <CONDITIE/RESTRICTIE> se referă la obiectul acțiunii, adică la elementul asupra căruia se acționează, și la mediul în care se desfășoară acțiunea ;

²⁾ <PARAMETRU> se referă la instrumentul acțiunii, adică la elementul cu care se acționează, în cazul de față fiind vorba de un produs informatic ;

³⁾ <REZULTAT POSIBIL> reprezintă unul din rezultatele ce pot apărea în urma acțiunii preconizate ; pentru a putea fi comparat cu obiectivul urmărit trebuie să fie exprimat în aceleasi unități de măsură cu acesta din urmă ;

⁴⁾ <SELECTIA OPERATIEI URMATOARE> exprimă cererea de efectuare a unei noi acțiuni, în funcție de rezultatul acțiunii curente ;

⁵⁾ Predicatul [ESTE FUNCTIE DE] (vezi tabela 1) se particularizează printr-o serie de condiții de forma :

<DACA> <REZULTAT POSIBIL 1> (SI) <REZULTAT OBTINUT 1> (SI)
 <PARAMETRU 1> (SI) <CONDITIE/RESTRICTIE 1> (SI) ... <ATUNCI>
 <COMANDA 1>

în care :

<REZULTAT POSIBIL 1> ::= <IPOTEZA> [ESTE (IN ACEST CAZ)] <VALOARE 1>
 <REZULTAT OBTINUT 1> ::= <REZULTAT> ESTE (IN ACEST CAZ)] <VALOARE 11>
 <PARAMETRU 1> ::= <PARAMETRU> [ESTE (IN ACEST CAZ)] <VALOARE 12>
 <CONDITIE/RESTRICTIE 1> ::= <CONDITIE/RESTRICTIE> [ESTE (IN ACEST CAZ)]
 <VALOARE 13>
 <COMANDA 1> ::= <SE> [MERGE LA]* <OPERATIE 2>.

Execuția acestei proceduri se face conform următorului algoritm : (vezi notațiile din tabela 3) :

PROGRAM PRINCIPAL :

pentru <SUBPROGRAM>

||Inlocuiește <OPERATIE x> cu <PRIMA OPERATIE> din <PROCEDURA>||

□

1. execută <SUBPROGRAM> □

dacă <MESAJ> = 'PROGRAM TERMINAT' atunci stop □

dacă <MESAJ> = 'CONTINUA PROGRAMUL' atunci

pentru <SUBPROGRAM>

||Înlocuiește <OPERATIE X> cu <OPERATIA URMATOARE> din <PROCEDURA>||

□
mergi la 1 □

□
dacă <MESAJ> = 'CONTINUARE IMPOZIBILA' atunci stop □

SUBPROGRAM :: =

||Execută <OPERATIE X> conform <DEFINITIE> și înregistrează <REZULTAT>, <PARAMETRU>, <CONDITIE/RESTRICTIE> legate de <OPERATIE X EXECUTATA>|| □

MESAJ 'CONTINUARE IMPOZIBILA' □

dacă <OBIECTIV> = <REZULTAT> atunci
MESAJ ← 'PROGRAM TERMINAT'
întoarce

□
execută <SELECTIA OPERATIEI URMATOARE> □
dacă <MESAJ> = 'CONTINUA PROGRAMUL' atunci
execută <FIXARE PARAMETRI>
execută <FIXARE CONDITII/RESTRICTII>

□
întoarce

Este interesant de notat faptul că atât programul principal cit și subprogramul de mai sus sint de fapt fraze în limbaj INTELEC, înregistrate în baza de date relațională (BDR) și legate la entitățile <PROGRAM PRINCIPAL>, respectiv <SUBPROGRAM>. Deoarece aceste fraze reprezintă *comenzi interne*, predicatele lor vor fi *predicate interne*, direct executabile de calculator. În afară de predicate, în cadrul frazelor se folosesc noțiuni care se referă la elementele din BDR și care sint și ele *noțiuni interne*.

Dăm mai jos, traducerea algoritmilor PROGRAM PRINCIPAL și SUBPROGRAM în limbaj INTELEC sub formă de texte purtind aceste nume.

<PROGRAM PRINCIPAL>

1. <SE> [INLOCUIESTE]* <OPERATIE>¹⁾ (DIN) <SUBPROGRAM> (CU) <PRIMA OPERATIE> (AL/A) <PROCEDURA>
2. <SE> [EXECUTA] * <SUBPROGRAM>
3. <DACA> <MESAJ> [ESTE EGAL CU]* <'PROGRAM TERMINAT'> <ATUNCI>
<SE> [OPRESTE] * <PROGRAM PRINCIPAL>
4. <DACA> <MESAJ> [ESTE EGAL CU]* <'CONTINUA PROGRAMUL'> <ATUNCI>
<SE> [INLOCUIEȘTE]* <OPERATIE>¹⁾ (DIN) <SUBPROGRAM> (CU)
<OPERATIA URMATOARE> (AL/A) <PROCEDURA> (SI) <SE> [MERGE LA]* (2) <FRAZA>* (AL/A) <PROGRAM PRINCIPAL>
5. <DACA> <MESAJ> [ESTE EGAL CU]* <'CONTINUARE IMPOZIBILA'> <ATUNCI>
<SE> [OPRESTE] * <PROGRAM PRINCIPAL>

<SUBPROGRAM>

1. <SE> [EXECUTA] * <DEFINITIE> (AL/A) <OPERATIE>
2. <SE> [INREGISTREAZA]* <OPERATIE>¹⁾ (LA) (EXECUTAT) <PROCEDURA>
3. <SE> [INREGISTREAZA]* (PL) <REZULTAT> (SI) (PL) <PARAMETRU> (SI)
(PL) <CONDITIE/RESTRICTIE> (LA) (EXECUTAT) <OPERATIE>



4. <SE> [ACTUALIZEAZA]* <'CONTINUARE IMPOZIBILA'> (IN) <MESAJ>
5. <SE> [EXECUTA]* <CONTROLUL REALIZARII OBIECTIVELOR>
6. <DACA> <MESAJ> [ESTE EGAL CU]* <'PROGRAM TERMINAT'> <ATUNCI>
<SE> [INTOARCE LA]* <PROGRAM PRINCIPAL>
7. <SE> [EXECUTA]* <SELECTIA OPERATIEI URMATOARE> (AL/A) <OPERATIE>
(AL/A) <PROCEDURA>
8. <DACA> <MESAJ> [ESTE EGAL CU]* <'CONTINUA PROGRAMUL'> <ATUNCI>
<SE> [EXECUTA]* <FIXARE PARAMETRI> (SI) <FIXARE CONDITII/RESTRICTII>

Pentru frazele de mai sus sunt valabile următoarele observații :

a. <PROGRAM PRINCIPAL> și <SUBPROGRAM> reprezintă entități în baza de date relațională și se regăsesc cu procedurile cunoscute (vezi capitolul 3).

b. Frazele numerotate cu 1 pînă la 5 în <PROGRAM PRINCIPAL> precum și cele numerotate cu 1 pînă la 8 în <SUBPROGRAM> trebuie parcuse *secrențial*, respectînd ordinea firească a lecturii.

c. Predicatul acestor fraze, marcate cu „*” au o anumită semnificație pentru calculator care, în urma identificării lor execută automat o serie de operații în baza de date. Astfel, predicatul [INLOCUIESTE]* (vezi tabela 1) arată că trebuie căutată entitatea-complement situată înaintea calificatorului (CU) și, oriunde este ea întîlnită în cadrul paragrafului precedent de (DIN), trebuie înlocuită cu entitatea situată după calificatorul (CU). În cazul primei fraze (notată cu 1) din <PROGRAM PRINCIPAL> această regulă se traduce prin identificarea <OPERATIE> și înlocuirea ei în cadrul <SUBPROGRAM> cu <PRIMA OPERATIE> din <PROCEDURA>. [EXECUTA]* (vezi tabela 1) indică parcurgerea tuturor frazelor înregistrate la entitatea-complement și apoi revenirea și citirea frazei următoare, ceea ce, în cazul frazei 2 din <PROGRAM PRINCIPAL> înseamnă parcurgerea (și executarea) tuturor frazelor din <SUBPROGRAM> după care se revine în <PROGRAM PRINCIPAL> ș.a.m.d.

d. Entitatea <OPERATIE>¹⁹ este entitatea *currentă* din <SUBPROGRAM>, cu alte cuvinte entitatea (particulară) înscrisă în momentul respectiv la <OPERATIE> aparținînd acestui paragraf. Ea poate fi oricînd înlocuită cu altă entitate (particulară) și atunci toate frazele înscrise în paragraful <SUBPROGRAM> referitoare la <OPERATIE> vor fi valabile pentru aceasta din urmă. De exemplu, <PRIMA OPERATIE>, definită în cadrul <PROCEDURA>, înlocuiește, pentru prima parcurgere a paragrafului <SUBPROGRAM> entitatea <OPERATIE>. Pentru revenirile ulterioare la acest paragraf entitatea <OPERATIE> va fi particularizată de fiecare dată prin parcurgerea paragrafului <SELECTIA OPERATIEI URMATOARE>.

e. Succesiunea operațiilor în cazul executării <PROCEDURA> într-un anume caz particular va fi înregistrată la <PROCEDURA> executată, sub forma unei fraze de tipul :

(EXECUTAT) <PROCEDURA> [SE COMPUNE DIN (SUCCEIUNE)] <OPERATIE 1>
(URMAT DE) <OPERATIE 2> (URMAT DE)...

Această frază va fi generată (și completată) automat la întîlnirea predicatului [INREGISTREAZA]* și constituie un istoric al execuției care poate fi exploarat ulterior în scopuri legate de analiză și perfecționare.

În mod analog, se vor completa valorile (particulare) ale <REZULTAT>, <PARAMETRU> și <CONDITIE/RESTRICTIE> conform frazei 3 din <SUBPROGRAM>.

f. Entitatea **⟨MESAJ⟩** va avea valori particulare după fiecare parcurgere a paragrafului **⟨SUBPROGRAM⟩** (executarea unei operații). Acestea vor fi marcate printr-o frază de tipul

⟨MESAJ⟩ [ESTE (ÎN ACEST CAZ)] ...

La întâlnirea predicatului **[ESTE EGAL CU]*** (vezi tabela 1) se compară subiectul cu complementul și în caz de egalitate se execută sirul de comenzi care urmează.

g. După parcurgerea ultimei fraze din **⟨SUBPROGRAM⟩**, în cazul, în care revenirea nu s-a făcut datorită unei comenzi anterioare, se revine automat în **⟨PROGRAM PRINCIPAL⟩** și se continuă parcurgerea acestuia din urmă.

h. Acțiunea se oprește la întâlnirea comenzi

⟨SE⟩ [OPRESTE]* ⟨PROGRAM PRINCIPAL⟩.

4.3. Aplicație în cazul utilizării pachetului de programe SOFTPLAN în întreprinderi

Să aplicăm acum cele de mai sus la pachetul de programe SOFTPLAN pentru conducerea operativă a producției în întreprinderile cu profil de construcții de mașini *).

Presupunem că **⟨PROCEDURA⟩** este particularizată în cazul nostru prin **⟨PROCEDURA DE OPTIMIZARE SOFTPLAN⟩**, la care sunt legate fraze care definesc: scopul, obiectivele, operațiile, rezultatele posibile, parametrii pachetului de programe, condițiile / restricțiile de aplicare precum și prima operație cu care începe aplicarea pachetului de programe. La fiecare din operații vom regăsi frazele de definiție, condițiile de selecție ale operației următoare, fixarea parametrilor și a condițiilor/restricțiilor pentru fiecare caz de selecție în parte. Vom avea, prin urmare :

⟨PROCEDURA DE OPTIMIZARE SOFTPLAN⟩

⟨SCOP⟩ [ESTE (IN ACEST CAZ)] ⟨ATINGERE⟩ (AL/A) ⟨OBIECTIV⟩ (PRIN)(MAXIMA) ⟨UTILIZARE⟩ (AL/A) ⟨MODUL SIM1B⟩ (SI) (AL/A) ⟨MODUL SIM2A⟩)**
⟨OBIECTIV⟩ [ESTE (IN ACEST CAZ)] ⟨SCADERE⟩ (AL/A) ⟨MAX⟩ (SUB) ⟨LIM⟩ (SI) ⟨OBTINERE⟩ (ACCEPTABIL) (PL) ⟨GOL DE INCARCARE⟩
⟨LIM⟩ [ESTE (ÎN ACEST CAZ)] VALOARE CONVENABIL ALEASĂ
⟨OPERATIE⟩ ⟨SE COPMUNE DIN (ANSAMBLU)] ⟨PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩ (SI) ⟨MODUL SIM1B⟩ (SI) ⟨MODUL SIM2A⟩ (SI) ⟨ACTUALIZARE MANUALĂ INTERMEDIARA⟩ (SI) ⟨AJUSTARE MANUALA FINALA⟩
⟨REZULTAT POSIBIL⟩ [ESTE (ÎN ACEST CAZ)] ⟨MAX⟩ (SI) ⟨GOL DE SARCINA⟩ (SI) ⟨SUPRAINCARCARI⟩
⟨PARAMETRU⟩ [ESTE (IN ACEST CAZ)] ⟨LIM⟩ (SI) ⟨DIV⟩ (SI) ⟨PRIO PE COMANDA⟩ (SI) ⟨ORDINE DE PRIORITATE A COMENZII⟩ (SI) ⟨TID⟩ (SI) ⟨TST⟩

*) Vezi descrierea pachetului în cadrul capitolului 6.

**) Modulele SIM1B și SIM2A reprezintă programe automate de optimizare (vezi cap. 6). Este indicată folosirea lor la maximum pentru a ușura munca omului în procesul de planificare/programare a încărcării capacitaților de producție.

⟨CONDITIE/RESTRICTIE⟩ [ESTE (ÎN ACEST CAZ)] ⟨FORMA CICLOGRAMELOR DE FABRICATIE⟩ (SI) ⟨NUMAR DE PRODUSE PE COMANDA⟩ (SI) ⟨RESTRICTII PENTRU COMENZILE CARE INCARCA PERIOADA IN CARE ARE LOC MAX⟩ (SI) ⟨RESTRICTII PENTRU CAPACITATEA PERIOADEI IN CARE ARE LOC MAX⟩

⟨PRIMA OPERATIE⟩ [ESTE (IN ACEST CAZ)] ⟨PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩

⟨PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩

⟨PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨DEFINITIE⟩ (SI) ⟨SELECTIA OPERATIEI URMATOARE⟩

⟨DEFINITIE⟩ [ESTE (ÎN ACEST CAZ)] ⟨DEFINITIE AL/A PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩

⟨SELECTIA OPERATIEI URMATOARE⟩ [ESTE (IN ACEST CAZ)] ⟨SELECTIA OPERATIEI URMATOARE AL/A PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩

⟨DEFINITIE AL/A PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩

⟨DEFINITIE AL/A PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨LISTARE BILANT PE POSTURI⟩ (SI) ⟨LISTARE INCARCARE PE COMENZI⟩ (SI) ⟨ACTUALIZARE MANUALA⟩

⟨ACTUALIZARE MANUALA⟩ [ESTE FUNCTIE DE] ⟨CICLURI DE FABRICATIE⟩

⟨DACA⟩ ⟨CICLURI DE FABRICATIE⟩ [PREZINTA IN MOD NEJUSTIFICAT INTRE-RUPERI] ⟨ATUNCI⟩ ⟨SE⟩ [DIVID IN MAI MULTE SUBCICLURI INDEPENDENTE]

⟨DACA⟩ ⟨CICLURI DE FABRICATIE⟩ [PREZINTA INCARCARI MICI LA INCEPUT SAU LA SFIRSI] ⟨ATUNCI⟩ ⟨SE⟩ [CALCULEAZA SAU SE NEGLIJEAZA SCURTIND CICLURILE]

⟨SELECTIA OPERATIEI URMATOARE AL/A PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE⟩

⟨SE⟩ [ACTUALIZEAZA]* ⟨MODUL SIM1B⟩ (IN) ⟨OPERATIA URMATOARE⟩

⟨SE⟩ [ACTUALIZEAZA]* ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩

⟨MODUL SIM1B⟩

⟨MODUL SIM1B⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨DEFINITIE⟩ (SI) ⟨SELECTIA OPERATIEI URMATOARE⟩

⟨DEFINITIE⟩ [ESTE (IN ACEST CAZ)] ⟨DEFINITIE AL/A MODUL SIM1B⟩

⟨SELECTIA OPERATIEI URMATOARE⟩ ESTE (ÎN ACEST CAZ)] ⟨SELECTIA OPERATIEI URMATOARE AL/A MODUL SIM1B⟩

⟨DEFINITIE AL/A MODUL SIM1B⟩

⟨SE⟩ [EXECUTA AUTOMAT]*) ⟨OPERATIE⟩

⟨OPERATIE⟩ [ESTE (IN ACEST CAZ)] ⟨DEPLASARE⟩ (AL/A) ⟨COMANDA⟩ (FARA) ⟨DIVIZARE⟩

*) Acest predicat indică faptul că execuția operației desemnată prin complement se efectuează în mod automat de către calculator prin aplicarea pachetului de programe. Rolul omului este de a supraveghea execuția și a înregistra rezultatele.

⟨CONDITIE DE ITERATIE REUSITA⟩ [ESTE (IN ACEST CAZ)] ⟨MAX⟩ (MAI MIC DECIT) (INITIAL) ⟨MAX⟩ (SAU) ⟨NUMAR DE MAX EGALE⟩ (MAI MIC DECIT) (INITIAL) ⟨NUMAR DE MAX EGALE⟩
 ⟨CRITERIU DE SELECTIE⟩ [ESTE (IN ACEST CAZ)] ⟨MAXIMUM⟩ (AL/A) ⟨SCADERE⟩ (AL/A) ⟨MEDIA REZERVEI DE INCARCARE RELATIVA⟩

⟨SELECTIA OPERATIEI URMATOARE AL/A MODUL SIM1B⟩

⟨DACA⟩ ⟨MAX⟩ [ESTE MAI MARE DECIT] ⟨LIM⟩ (SI) ⟨GOLURI⟩ [ESTE APROXIMATIV EGAL CU] ⟨SUPRAINCARCARI⟩ ⟨ATUNCI⟩ ⟨SE⟩ [ACTUALIZEAZA]* ⟨MODUL SIM2A⟩ (IN) ⟨OPERATIA URMATOARE⟩ (SI) ⟨SE⟩ [ACTUALIZEAZA]* ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩

⟨MODUL SIM2A⟩

⟨MODUL SIM2A⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨DEFINITIE⟩ (SI) ⟨SELECTIA OPERATIEI URMATOARE⟩
 ⟨DEFINITIE⟩ [ESTE IN ACEST CAZ] ⟨DEFINITIE AL/A MODUL SIM2A⟩
 ⟨SELECTIA OPERATIEI URMATOARE⟩ [ESTE (IN ACEST CAZ)] ⟨SELECTIA OPERATIEI URMATOARE AL/A MODUL SIM2A⟩

⟨DEFINITIE AL/A MODUL SIM2A⟩

⟨SE⟩ [EXECUTA AUTOMAT]* ⟨OPERATIE⟩
 ⟨OPERATIE⟩ [ESTE (IN ACEST CAZ)] ⟨DEPLASARE⟩ (AL/A) ⟨COMANDA⟩ (CU) ⟨DIVIZARE⟩
 ⟨COMANDA⟩ [ESTE (IN ACEST CAZ)] (CU PRIORITATEA CEA MAI MICA) ⟨COMANDA⟩
 ⟨CONDITIE DE ITERATIE REUSITA⟩ [ESTE (IN ACEST CAZ)] ⟨MAX⟩ (MAI MIC DECIT) (INITIAL) ⟨MAX⟩ (SAU) ⟨NUMAR DE MAX EGALE⟩ (MAI MIC DECIT) (INITIAL) ⟨NUMAR DE MAX EGALE⟩
 ⟨CRITERIUL DE SELECTIE⟩ [ESTE (IN ACEST CAZ)] (PRIMA) ⟨ITERATIE REUSITA⟩

⟨SELECTIA OPERATIEI URMATOARE AL/A MODUL SIM2A⟩

⟨DACA⟩ ⟨MAX⟩ [ESTE MAI MIC SAU EGAL CU] ⟨LIM⟩ (SI)
 ⟨LIM⟩ [ESTE MAI MARE DECIT] ⟨VALOARE CONVENABIL ALEASA⟩
 ⟨ATUNCI⟩ ⟨SE⟩ [ACTUALIZEAZA]* ⟨MODUL SIM2A⟩ (IN)
 ⟨OPERATIA URMATOARE⟩ (SI) ⟨SE⟩ [ACTUALIZEAZA]*
 ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩
 ⟨DACA⟩ ⟨MAX⟩ [ESTE MAI MARE DECIT] ⟨LIM⟩ ⟨ATUNCI⟩
 ⟨SE⟩ [ACTUALIZEAZA]* ⟨ACTUALIZARE MANUALĂ INTERMEDIARĂ⟩ (IN)
 ⟨OPERATIE URMATOARE⟩ (SI)
 ⟨SE⟩ [ACTUALIZEAZA]* ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩

⟨ACTUALIZARE MANUALA INTERMEDIARA⟩

⟨ACTUALIZARE MANUALA INTERMEDIARA⟩ [SE COMPUNE DIN (ANSAMBLU)]
 ⟨DEFINITIE⟩ (SI) ⟨SELECTIA OPERATIEI URMATOARE⟩
 ⟨DEFINITIE⟩ [ESTE (IN ACEST CAZ)] ⟨DEFINITIE AL/A ACTUALIZARE MANUALA INTERMEDIARA⟩

⟨SELECTIA OPERATIEI URMATOARE⟩ [ESTE (IN ACEST CAZ)] ⟨SELECTIA OPERATIEI URMATOARE AL/A ACTUALIZARE MANUALA INTERMEDIARA⟩

⟨DEFINITIE AL/A ACTUALIZARE MANUALA INTERMEDIARA⟩

⟨DACA⟩ ⟨RESTRICTII PENTRU COMENZILE CARE INCARCA PERIOADA IN CARE ARE LOC MAX⟩ [ESTE PREA MARE] ⟨ATUNCI⟩ ⟨SE⟩ [DEPLASEAZA MANUAL INCARCARILE DIN PERIOADA IN CARE ARE LOC MAX] (SAU) ⟨SE⟩ [RIDICA RESTRICTIILE PENTRU DEPLASAREA SI DIVIZAREA COMENZILOR CARE INCARCA PERIOADA IN CARE ARE LOC MAX]

⟨DACA⟩ ⟨RESTRICTII PENTRU CAPACITATEA PERIOADEI IN CARE ARE LOC MAX [ESTE PREA MARE] (SI) ⟨CAPACITATEA POSTULUI IN PERIOADA MAX⟩ [SE POATE MARI] ⟨ATUNCI⟩ ⟨SE⟩ [MAJOREAZA] ⟨CAPACITATEA POSTULUI IN PERIOADA MAX⟩

⟨SELECTIA OPERATIEI URMATOARE AL/A ACTUALIZARE MANUALA INTERMEDIARA⟩

⟨DACA⟩ [S-AU RIDICAT O PARTE DIN RESTRICTIILE PENTRU COMENZI SI CAPACITATI CONFORM DEFINITIEI] ⟨ATUNCI⟩ ⟨SE⟩ [ACTUALIZEAZA]*

⟨MODUL SIM2A⟩ (IN) ⟨OPERATIA URMATOARE⟩ (SI)

⟨SE⟩ [ACTUALIZEAZA]* ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩

⟨DACA⟩ [NU S-AU RIDICAT NICI UN FEL DE RESTRICTII REFERITOARE LA COMENZI SI CAPACITATI, SITUATIA RAMINE IDENTICA CU ACEEA DE LA SFIRSIUL FAZEI PRECEDENTE] ⟨ATUNCI⟩ ⟨SE⟩ [ACTUALIZEAZA]*
 ⟨AJUSTARE MANUALA FINALA⟩ (IN) ⟨OPERATIA URMATOARE⟩ (SI)
 ⟨SE⟩ [ACTUALIZEAZA]* ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩

⟨AJUSTARE MANUALA FINALA⟩

⟨AJUSTARE MANUALA FINALA⟩ [SE COMPUNE DIN (ANSAMBLU)] ⟨DEFINITIE (SI) ⟨SELECTIA OPERATIEI URMATOARE⟩

⟨DEFINITIE⟩ [ESTE (IN ACEST CAZ)] ⟨DEFINITIE AL/A AJUSTARE MANUALA FINALA⟩

⟨SELECTIA OPERATIEI URMATOARE⟩ [ESTE (IN ACEST CAZ)] ⟨SELECTIA OPERATIEI URMATOARE AL/A AJUSTARE MANUALA FINALA⟩

⟨DEFINITIE AL/A AJUSTARE MANUALA FINALA⟩

⟨SE⟩ [DISTRIBUIE MANUAL SUPRAINCARCARILE MARI] (PE) (PL) ⟨POST DE LUCRU ECHIPAMENT⟩ (SAU/SI)

⟨SE⟩ [IAU MASURI] (PENTRU) ⟨EXECUTAREA OPERATIILOR RESPECTIVE LA ALTE ATELIERE, FABRICI SAU INTreprinderi⟩ (SAU/SI)

⟨SE⟩ [IAU MASURI ORGANIZATORICE] (PENTRU) ⟨INCEPEREA MAI DEVREME A FABRICATIEI, OBTINEREA UNOR DECALARI DE TERMENE ETC.⟩ (SAU/SI)

⟨SE⟩ [REDUCE CORESPUNZATOR] ⟨NUMARUL PRODUSELOR DE FABRICAT⟩

〈SELECTIA OPERATIEI URMATOARE AL/A AJUSTARE MANUALĂ FINALĂ〉*

〈FIXARE PARAMETRI〉 **)

〈DACA〉 〈OPERATIA URMATOARE〉 [ESTE EGAL CU] * 〈MODUL SIM1B〉
〈ATUNCI〉 〈SE〉 [ACTUALIZEAZĂ]* 〈LIM〉 (SI) 〈TID〉 (SI) 〈TST〉 (SI)
〈ORDINEA DE PRIORITATE〉 (IN) 〈PARAMETRU〉
 (SI) 〈SE〉 [STABILESTE]* 〈LIM〉 (LA) 〈VALOARE CONVENABIL ALEASĂ¹⁾
 (SI) 〈SE〉 [FIXEAZĂ]* 〈TID〉 (AL/A) (PL) 〈COMANDA〉 (SI) 〈SE〉 [FIXEAZĂ]*
 〈TST〉 (AL/A) (PL) 〈COMANDA〉 (SI) 〈SE〉 [FIXEAZA] * 〈ORDINEA DE
 PRIORITATE〉 (AL/A) (PL) 〈COMANDA〉
 〈DACA〉 〈OPERATIA URMATOARE〉 [ESTE EGAL CU] * 〈MODUL SIM2A〉 〈ATUNCI〉
 〈SE〉 [ACTUALIZEAZĂ] * 〈LIM〉 (SI) 〈DIV〉 (SI) 〈PRIO PE COMANDA〉
 (IN) 〈PARAMETRU〉 〈SE〉 [STABILESTE]* 〈LIM〉 (LA) 〈VALOARE1 CON-
 VENABIL ALEASA²⁾ (SI) 〈SE〉 [STABILESTE]* 〈DIV〉 (LA) 〈VALOARE2
 CONVENABIL ALEASA³⁾ (SI) 〈SE〉 [FIXEAZA] * 〈PRIO PE COMANDA⁴⁾ (AL/A) (PL) 〈COMANDA〉

în care :

¹⁾ 〈VALOARE CONVENABIL ALEASA〉 este valoarea maximă a supra-
 sarcinii restante care nu deranjează pe utilizator, putind fi compensată
 prin măsuri organizatorice locale.

²⁾ 〈VALOARE1 CONVENABIL ALEASA〉 este funcție de 〈MAX〉 și de du-
 rata „reprizei” pe calculator, aleasă în aşa fel încit, din motive de sigu-
 ranță și de ușurință a programării, să nu depășească 2–3 ore de funcțio-
 nare continuu.

〈VALOARE1 CONVENABIL ALEASA〉 este mai mare sau cel mult egală cu
 〈VALOARE CONVENABIL ALEASA〉.

³⁾ 〈VALOARE2 CONVENABIL ALEASA〉 este funcție de numărul maxim
 de divizări pentru fiecare comandă, ales de utilizator.

⁴⁾ 〈PRIO PE COMANDA〉 este egal cu 0 pentru comanda care nu trebuie
 mișcată și nu are voie să fie divizată, este egal cu 1 pentru comanda care
 poate fi deplasată în intervalul TID–TST dar nu trebuie divizată și este
 egal cu 2 pentru comanda care poate fi mișcată și divizată după necesită-
 ţile algoritmului de optimizare.

⁵⁾ 〈ORDINEA DE PRIORITATE〉 este ordinea în care sunt inserate comen-
 zile în lista de comenzi ce trebuie programate; primele comenzi sunt cel
 mai mult devansate în timp ce ultimele sunt cel mai mult decalate în urma
 optimizării.

〈FIXARE CONDITII/RESTRICTII〉 ***)

〈DACA〉 〈OPERATIA URMATOARE〉 [ESTE EGAL CU]* 〈ACTUALIZARE MANUALA
 INTERMEDIARA〉
 〈ATUNCI〉 〈SE〉 [ACTUALIZEAZĂ]* 〈FORMA CICLOGRAMELOR DE FABRICATIE〉
 (SI) 〈NUMAR DE PRODUSE PE COMANDA〉 (SI) 〈RESTRICTII PENTRU CO-
 MENZILE CARE INCARCA PERIOADA IN CARE ARE LOC MAX〉 (SI) 〈RES-
 TRICTII PENTRU CAPACITATEA PERIOADEI IN CARE ARE LOC MAX〉 (IN)
 〈CONDITIE/RESTRICTIE〉.

*) Fiind vorba de ultima operație, acest text rămîne vid.

**) Valabil pentru toate operațiile.

***) Valabil pentru toate operațiile.

〈CONTROLUL REALIZARII OBIECTIVELOR〉^{*)}

〈DACA〉 [〈MAX〉 ESTE MAI MIC SAU EGAL CU] 〈LIM〉 (SI) 〈LIM〉 [ESTE EGAL CU] 〈VALOARE CONVENABIL ALEASA〉
 〈ATUNCI〉 (SE) [ACTUALIZEAZĂ]^{*} 〈'PROGRAM TERMINAT'〉 (IN) 〈MESAJ〉

Să trecem acum la executarea programului.

Formulăm cererea în modul următor :

Comenzi și răspunsuri la consola terminalului de teletransmisie	Note explicative
-u: EXEC PROGRAM ?	Se cheamă programul monitor care supraveghează dialogul.
-c: NUME ?	Calculatorul cere numele programului de executat.
-u: PROGRAM PRINCIPAL** PROCEDURA = PROCEDURA DE OPTIMIZARE SOFTPLAN	Se furnizează numele programului indicind totodată prin semnul „**” că în cadrul acestuia 〈PROCEDURA〉 se înlocuiește cu 〈PROCEDURA DE OPTIMIZARE SOFTPLAN〉. Se generează fraza următoare : 〈PROCEDURA〉 ESTE (IN ACEST CAZ) 〈PROCEDURA DE OPTIMIZARE SOFTPLAN〉.
<p>Urmează execuția programului care, pentru ușurință urmăririi, va fi parcurs pas cu pas. Vom nota cu: <i>Pn</i> execuția frazei <i>n</i> din 〈PROGRAM PRINCIPAL〉 și cu <i>Sm</i> execuția frazei <i>m</i> din 〈SUBPROGRAM〉.</p> <p><i>P 1:</i> Se înlocuiește în 〈SUBPROGRAM〉 entitatea 〈OPERATIE〉 cu 〈PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE〉</p> <p><i>P 2:</i> Se trece la executarea 〈SUBPROGRAM〉</p> <p><i>S 1:</i> Se citesc și se execută frazele care definesc 〈PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE〉</p> <p><i>S 2:</i> Se generează fraza : (EXECUTAT) 〈PROCEDURA DE OPTIMIZARE SOFTPLAN〉 SE [COMPUNE DIN (SUCCESIUNE)] 〈PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE〉</p> <p><i>S 3:</i> Se înscriu în BDR frazele referitoare la rezultatele obținute : (EXECUTAT) 〈PREGATIRE MANUALA INITIALA PENTRU OPTIMIZARE〉 [SE COMPUNE DIN (SUCCESIUNE)] 〈REZULTAT〉 (URMAT DE) 〈PARAMETRU〉 (URMAT DE) 〈CONDITIE/RESTRICTIE〉 〈REZULTAT〉] ESTE (IN ACEST CAZ) 〈DIVIZARE〉 (AL/A) 〈COM010〉 (AL/A) 〈COMANDA〉 (IN) 〈COM011〉 (AL/A) 〈COMANDA〉 (SI) 〈COM012〉 (AL/A) 〈COMANDA〉 〈PARAMETRU〉 ESTE (IN ACEST CAZ) ... 〈CONDITIE/RESTRICTIE〉 ESTE ([IN ACEST CAZ]) ...</p> <p><i>S 4:</i> Se generează fraza : 〈MESAJ〉 [ESTE (IN ACEST CAZ)] 〈'CONTINUARE IMPOZIBILA'〉.</p> <p><i>S 5:</i> Se execută 〈CONTROLUL REALIZARII OBIECTIVELOR〉. Deoarece 〈MAX〉 este pentru inceput, mult mai mare ca 〈LIM〉, obiectivul este nerealizat și mesajul rămîne ca mai înainte.</p>	

^{*)} Valabil pentru toate operațiile

S 6: Condiția nu este îndeplinită. Se trece la fraza următoare.

S 7: Se execută <SELECTIA OPERATIEI URMATOARE> aparținind operației <PREGATIREA MANUALA INITIALA PENTRU OPTIMIZARE>.

Se generează fraza :

<OPERAȚIA URMATOARE> [ESTE (IN ACEST CAZ)] <MODUL SIM1B> și se modifică <MESAJ> :

<MESAJ> [ESTE (IN ACEST CAZ)] <'CONTINUA PROGRAMUL'>.

S 8: Condiția este îndeplinită. Se execută <FIXARE PARAMETRI>.

Calculatorul cere precizarea entităților <VALOARE CONVENABIL ALEASA> precum și <TID>, <TST> și <ORDINEA DE PRIORITATE> pentru toate comenziile.

c: VALOAREA CONVENABIL ALEASA ? Se cere fixarea valorii <LIM>.

u: 500 Utilizatorul răspunde indicind valoarea 500.

c: COMANDA ?

Se cere prima comandă în lista de priorități.

u: COM010

Utilizatorul răspunde indicind comanda COM010.

c: TID ?

Se cere timpul de inceput cel mai devreme TID.

u: 1

Utilizatorul răspunde indicind prima perioadă.

c: TST ?

Se cere timpul de sfîrșit cel mai tîrziu TST.

u: 12

Utilizatorul răspunde indicind perioada a 12-a.

c: COMANDA ?

Se trece la comanda următoare.

u: COM011

În urma dialogului se generează frazele :

<LIM> [ESTE (IN ACEST CAZ)] <500>

<COMANDA> [SE COMPUNE DIN (ANSAMBLU)] <COM010> (și) <COM011> (și) ...

<TID> (AL/A) <COM010> [ESTE (IN ACEST CAZ)] <1>

<TST> (AL/A) <COM010> [ESTE (IN ACEST CAZ)] <12>

Urmează întoarcerea la <PROGRAM PRINCIPAL> și executarea frazei următoare.

P 3: Condiția nu este îndeplinită. Se trece la fraza următoare.

P 4: Condiția este îndeplinită. Se înlocuiește <OPERATIE> din <SUBPROGRAM> cu <MODUL SIM1B>. Se continuă, conform indicației, cu fraza 2 din <PROGRAM PRINCIPAL>.

P 2: Se execută <SUBPROGRAM>.

S 1: Se citesc și se execută frazele care definesc <MODUL SIM1B> – este vorba de o execuție automată, pe calculator, în urma cărora se constată o serie de rezultate.

S 2: Se completează fraza.

(EXECUTAT) <PROCEDURA DE OPTIMIZARE SOFTPLAN> [ESTE (IN ACEST CAZ)] <PREGATIREA MANUALA INITIALA PENTRU OPTIMIZARE> (URMAT DE) <MODUL SIM1B>.

S 3: <REZULTAT> [ESTE (IN ACEST CAZ)] <MAX> (MĂI MARE DECIT) <LIM>.

S 4: <MESAJ> [ESTE (IN ACEST CAZ)] <'CONTINUARE IMPOZIBILA'>.



S 5: Obiectivul este încă nerealizat și se continuă programul.

S 6: Condiție neinndeplinită. Se trece la fraza următoare.

S 7: Se execută <SELECTIA OPERATIEI URMATOARE> aparținind operației <MODUL SIM1B>.

Se actualizează frazele :

<OPERATIE URMATOARE> [ESTE (ÎN ACEST CAZ)] <MODUL SIM2A>

<MESAJ> [ESTE (ÎN ACEST CAZ)] <'CONTINUA PROGRAMUL'>

ș.a.m.d.

4.4. Introducerea unor elemente de inteligență artificială în planificarea deciziilor privind alegerea fazelor de optimizare

Programul de mai sus continuă să lucreze pînă cînd se intilnește mesajul <'PROGRAM TERMINAT'> ceea ce arată că s-au obținut rezultatele secontate și programarea este satisfăcătoare. Dar, în mod cu totul neplăcut, programul se oprește și atunci cînd intilnește mesajul <'CONTINUARE IMPOSSIBILA'>. Luerul acesta se întimplă în cazul în care lipsesc informațiile necesare pentru a face selecția operației următoare sau, ceea ce este echivalent, rezultatele obținute împreună cu parametrii și condițiile/restricții în care s-a lucrat *nu se încadrează în nici una din situațiile prevăzute și pentru care există o decizie de continuare înscrișă în baza de date*.

Avem de-a face, așa dar, în acest din urmă caz, cu o *situatie inedită* și nu cu o decizie preprogramată — un caz tipic de *intervenție a inteligenței artificiale* pentru rezolvarea problemei. Cum lucrează modelul nostru în această situație?

Desigur, dacă cel care exploatează modelul ar cunoaște foarte bine construcția pachetului de programe, atunci, printr-o serie de *întrebări successive* adresate sistemului de calcul, folosind combinat informarea directă cu aceea indirectă (vezi paragraful 3.6b), ar putea găsi singur o continuare satisfăcătoare pentru situația în care se află. Ce se întimplă însă cu utilizatorii mai puțin avizați? Modelul nu i-ar putea ajuta cumva să iasă din impas?

De fapt, cum ar proceda un cunoscător în materie? Prima să grijă ar fi să vadă *care obiective* nu au fost încă satisfăcute. Apoi, ar căuta să afle *ce rezultate, parametri, condiții/restricții* sau combinații ale acestora au apărut în plus față de cazurile deja știute și inscrise în baza de date. Ar urma să încerce apoi să *încadreze* noua situație în unul din cazurile cunoscute, dacă, bineînțeles, diferențele ar fi neesențiale. Dacă ar constata însă diferențe mari, ar fi necesară o *analiză mai profundată a modului de funcționare* a pachetului de programe. În această analiză ar intra desigur și o serie de *simulări* ale comportamentului pachetului de programe urmate de *aprecieri* ale rezultatelor obținute, utilizatotul optind în final pentru continuarea care i-ar oferi cele mai mari șanse de reușită.

Pe baza acestui posibil comportament, să încercăm să construim, folosind modelul INTELEC, o rutină de „salvare” valabilă ori de cîte ori mesajul ‘CONTINUARE IMPOSSIBILA’ indică o situație de impas în programul nostru. Dăm mai jos această rutină.

SALVARE :: =

|| Listeașă : <OBIECTIV> ||
 || Identifică <OPERATIE X> = ultima operație executată ||
 pentru (EXECUTAT) <OPERATIE X>
 || listeașă : <REZULTAT>, <PARAMETRU>, <CONDITIE/RESTRICTIE> ||

 pentru <OPERATIE X>
 || listeașă : <DEFINITIE>, <SELECTIA OPERATIEI URMATOARE> ||

 || Înregistrează noua selecție la paragraful : <SELECTIA OPERATIEI
 URMATOARE> ||*
 mergi la <SE> [EXECUTA]* <SELECTIA OPERATIEI URMATOARE> în
 <SUBPROGRAM>

*) Aici intervine utilizatorul care, în urma raționamentului efectuat
 pe baza informațiilor primite, introduce în BDR un nou criteriu de selecție
 corespunzător situației în care se află. Criteriul ar putea avea, de exemplu,
 următoarea formă :

<DACA> <REZULTAT>...
 (SI) <PARAMETRU>...
 (SI) <CONDITIE/RESTRICTIE>...
 <ATUNCI> <SE> [ACTUALIZEAZA]* <OPERATIE X> (IN) <OPERATIA
 URMATOARE> (SI) <SE> [ACTUALIZEAZA]* <'CONTINUA PROGRAMUL'>
 (IN) <MESAJ>

Traducerea în limbaj INTELEC a rutinei de mai sus ar fi următoarea :

<SALVARE>

1. <SE> [AFISEAZA]* <OBIECTIV> (AL/A) <PROCEDURA>
2. <SE> [AFISEAZA]* <REZULTAT> (AL/A) (EXECUTAT) <ULTIMA OPERATIE>
 (SI) <PARAMETRU> (AL/A) (EXECUTAT) <ULTIMA OPERATIE> (SI)
 <CONDITIE/RESTRICTIE> (AL/A) (EXECUTAT) <ULTIMA OPERATIE>.
3. <SE> [AFISEAZA]* <DEFINITIE> (AL A) <ULTIMA OPERATIE> (SI) <SELEC-
 TIA OPERATIEI URMATOARE> (AL/A) <ULTIMA OPERATIE>.
4. <SE> [AFISEAZA]* <'NOUA SELECTIE ?'>.
5. <SE> [ASTEAPTA RASPUNS]*.

Iar folosirea ei în programele anterioare s-ar face după înlocuirea
 ultimei fraze (5) din <PROGRAM PRINCIPAL> cu o altă frază, de exemplu :

<PROGRAM PRINCIPAL>

5. <DACA> <MESAJ> [ESTE EGAL CU]* <'CONTINUARE IMPOSSIBILA'>
 <ATUNCI> <SE> [INLOCUIESTE]* <ULTIMA OPERATIE> (DIN) <SALVARE>
 (CU) <ULTIMA><OPERATIE> (AL/A) (EXECUTAT) <PROCEDURA> (SI)
 <SE> [EXECUTA]* <SALVARE> (SI) <SE> [MERGE LA]* (5)
 <FRAZA>* (AL/A) <SUBPROGRAM>.

Acum, pentru orice situație neprevăzută, utilizatorul va primi informațiile necesare, va fi obligat să alcătuiască un criteriu de selecție care să acopere cazul particular întâlnit iar programul nu va avea nici un motiv să se oprească înainte de atingerea țelului final.



Reluind raționamentul efectuat anterior, pentru „salvarea” situației ar mai exista și o a doua soluție: *calculatorul* să aleagă dintre continuările posibile pe aceea care oferă cele mai mari șanse de reușită și să continue programul. Acest lucru ar avea însă inconvenientul ca în lipsa unor criterii eficiente de selecție să prelungească mult durata obținerii unui rezultat satisfăcător. Dacă avem în vedere faptul că unii dintre utilizatori sunt capabili să aleagă o continuare mult mai bună pe baza experienței proprii și efectuând unele *inferențe*^{*)} greu de formalizat, metoda alegării automate ne apare ca fiind de cele mai multe ori inutilă. Dacă ne gîndim însă la utilizarea pachetelor de programe în întreprinderi, de către un personal mai puțin instruit și, cel puțin pentru început, lipsit de experiență, această formulă a deciziei pe calculator începe să devină interesantă.

Să vedem, prin urmare, cum ar arăta o rutină care ar încerca să execute în mod automat selecția operației care urmează, neavînd înregistrat în prealabil în baza de date o comandă expresă pentru această operație. Vom denumi această rutină DECIZIE.

DECIZIE ::=

||Identifică <OPERATIE X> = ultima operație executată||
 ||SEPARA SI EVALUEAZA¹⁾ fiecare criteriu de selecție din <SELECTIA OPERATIEI URMATOARE> (AL/A) <OPERATIE X> față de <REZULTAT>, <PARAMETRU> și <CONDITIE/RESTRICTIE> (AL/A) (EXECUTAT) <OPERATIE X>
	ALEGE²⁾ comanda de continuare care întrunește punetajul maxim	
	Actualizează <OPERATIA URMATOARE> (AL/A) <PROCEDURA>	
	Actualizează <MESAJ> cu <'CONTINUA PROGRAMUL'>	
 mergi la <SE> [EXECUTA]^{*} <FIXARE PARAMETRI> (SI) <FIXARE CONDITII/RESTRICTII> în <SUBPROGRAM>

în care :

¹⁾ SEPARA SI EVALUEAZA este un predicat intern care efectuează compararea fiecărui element care apare într-un criteriu de selecție al unei operații cu elementele corespunzătoare aparținând aceleiași operații execute. În cazul nostru, se compară pe rînd <REZULTAT> care apare în <SELECTIA OPERATIEI URMATOARE> (AL/A) <OPERATIE X> CU <REZULTAT> (AL/A) (EXECUTAT) <OPERATIE X>, <PARAMETRU> care apare în <SELECTIA OPERATIEI URMATOARE> (AL/A) <OPERATIE X>, cu <PARAMETRU> (AL/A) (EXECUTAT) <OPERATIE X> și, respectiv, <CONDITIE/RESTRICTIE> care apare în <SELECTIA OPERATIEI URMATOARE> cu <CONDITIE/RESTRICTIE> (AL/A) <OPERATIE X>.

^{*)} Numim *inferență* o operație logică prin care se admite o propoziție în virtutea legăturii sale cu alte propoziții considerate adevărate (*MicroRobert*, SNL Le Robert, 1971, p. 562.).

Pentru fiecare coïncidență, se acordă un *punctaj* (numărul de puncte acordat poate fi diferit în funcție de importanța criteriului). Evident, pentru elementele care nu coincid nu se acordă nici un punct. La sfîrșitul acestei operații, *fiecare frază* de selecție din <SELECTIA OPERATIEI URMATOARE> (AL/A) <OPERATIE X> va fi cotată cu un număr de puncte.

²⁾ ALEGE este un predicat intern care, utilizat după SEPARA SI EVALUEAZA, are rolul de a desemna comanda care întrunește punctajul maxim și care va fi aleasă pentru continuarea programului, ca fiind *cea mai apropiată* de condițiile noii situații de selecție. Această comandă se inscrie în <SELECTIA OPERATIEI URMATOARE>.

Algoritmul de mai sus se poate traduce și el într-o secvență de fraze în limbaj INTELEC :

<DECIZIE>

1. <SE> [SEPARA SI EVALUEAZA]* (FIECARE) <FRAZA>* (AL/A) <SELECTIA OPERATIEI URMATOARE> (AL/A) <ULTIMA OPERATIE>.
2. <SE> [ALEGE]* (DE CONTINUARE) <FRAZA>* (AL/A) <SELECTIA OPERATIEI URMATOARE> (AL/A) <ULTIMA OPERATIE>

Folosirea acestei rutine în programele anterioare se poate face ca și în cazul precedent, după înlocuirea frazei 5 din <PROGRAM PRINCIPAL>. De astă dată, fraza de înlocuire este următoarea :

<PROGRAM PRINCIPAL>

5. <DACA> <MESAJ> [ESTE EGAL CU]* <'CONTINUARE IMPOZIBILA'> <ATUNCI> <SE> [INLOCUIESTE]* <ULTIMA OPERATIE> (DIN) <DECIZIE> (CU) (ULTIMA) <OPERATIE> (AL/A) (EXECUTAT) <PROCEDURA> (SI) <SE> [EXECUTA]* <DECIZIE> (SI) <SE> [MERGE LA]* (7) <FRAZA>* (AL/A) <SUBPROGRAM>.

Este posibilă și combinarea celor două soluții de mai sus, prin întreruperea programului și întrebarea utilizatorului :

c: CE SOLUTIE PREFERATI? (0 = ALEGETI DUMNEAVOAESTRA INSIVA OPERATIA URMATOARE; 1 = CALCULATORUL ALEGE UNA DIN CONTINUARILE INSCRISE PINA IN MOMENTUL DE FATA IN BAZA DE DATE).

La răspunsul „0” se continuă programul cu <SALVARE> iar la răspunsul „1” cu <DECIZIE>, conform regulilor menționate anterior.



Este interesant de remarcat faptul că în cazul aplicării celei de-a două soluții, a DECIZIEI efectuate de calculator, este posibilă și o *îmbinătățire a operației de selecție*, după un număr oarecare de experiențe efectuate în acest sens.

Într-adevăr, având în vedere că fiecare selecție efectuată de calculator este înregistrată în baza de date la entitatea <EXECUTAT> <PROCEDURA> la care se leagă toate operațiile executate printr-o frază de tipul :

(EXECUTAT) <PROCEDURA> [SE COMPUNE DIN (SUCCESIUNE)]
 <OPERATIE 1> (URMAT DE) <OPERATIE 2> ...

iar rezultatele, parametrii și condițiile/restricții în care sunt efectuate aceste operații sunt înregistrate la entitatea respectivă prin fraze de tipul: **〈OPERAȚIE X〉 (AL/A) (EXECUTAT) 〈PROCEDURA〉 [SE COMPUNE DIN (SUCCESIUNE)] 〈REZULTAT〉 (URMAT DE) 〈PARAMETRU〉 (URMAT DE) 〈CONDITIE/RESTRICTIE〉**

〈REZULTAT〉 [ESTE (IN ACEST CAZ)] ...

〈PARAMETRU〉 [ESTE (IN ACEST CAZ)] ...

〈CONDITIE/RESTRICTIE〉 [ESTE (IN ACEST CAZ)] ...

este relativ simplu de analizat drumul parcurs de calculator la fiecare utilizare a programului PROCEDURA. În concluzie se poate vedea care dintre soluțiile adoptate a condus mai repede la rezultat și se pot prefera pe viitor aceste soluții – mai rapide, mai economice.

Acet gen de analiză se poate face fie manual, fie automat. Automatizarea poate fi efectuată însă numai dacă se formulează criterii precise de evaluare a variantelor, de exemplu: număr minim de operații parcuse, pondere maximă a operațiilor efectuate automat de calculator față de totalul operațiilor, minimul duratei totale de execuție etc. Aprecierea globală a acestor variante este de fapt o problemă de *decizie multidimensională* (vezi capitolul 2) și poate fi tratată ca atare.

În ipoteza în care ar fi posibilă formularea criteriilor de evaluare a variantelor de continuare, rutina de „autoperfecționare” a calculatorului – să o numim INVATARE – ar putea fi practicată după un număr oarecare de utilizări ale pachetului de programe asistat de modelul INTELEC și ar putea îmbunătăți considerabil sistemul de decizii automate, ducind la creșterea performanțelor pachetului de programe și a gradului său de adaptabilitate pentru diverse întreprinderi. Dăm mai jos algoritmul rutinei INVATARE folosit ca bază pentru transformarea modelului INTELEC într-un *model autoinstruibil*.

INVATARE :: =

pentru toate ॥(EXECUTAT) 〈PROCEDURA〉||

*||APRECIAZA¹⁾ fiecare 〈OPERAȚIE〉 după CRITERIUL²⁾1, CRITERIUL²⁾2,
CRITERIUL³⁾3, ... ||*

||ALEGE⁵⁾ cea mai convenabilă variantă||

*||INSCRIE⁶⁾ cea mai convenabilă variantă în 〈SELECTIA OPERATIEI
URMATOARE〉 pentru fiecare 〈OPERAȚIE〉 parcursă||*

în care :

¹⁾ APRECIAZA este predicatul care analizează fiecare variantă de 〈PROCEDURA〉 executată sub aspectul criteriilor menționate: CRITERIUL₁, CRITERIUL₂, CRITERIUL₃, ... Aprecierea variantelor *i* din punctul de vedere al fiecărui criteriu *j* se traduce prin acordarea unei note cuprinse între 0 și 1 fiecărei variante, cu 0 fiind notată varianta cea mai defavorabilă iar cu 1 varianta cea mai favorabilă. Se obține în final $A = (a_{ij})$, $i = 1, \dots, m$, $j = 1, \dots, n$, matricea aprecierilor normalizate monocriteriale [11, 12, 36].

De asemenea, fiecărui criteriu *j* i se acordă o pondere *k_j*, proporțională cu importanța sa față de celelalte criterii, în așa fel încât $\sum_{j=1}^n k_j = 1$.

²⁾ CRITERIUL₁ este numărul (minim) de operații parcuse pînă la obținerea rezultatului final.

³⁾ CRITERIUL2 este ponderea (maximă) a operațiilor efectuate automat de calculator față de totalul operațiilor parcuse pînă la obținerea rezultatului final.

⁴⁾ CRITERIUL3 este durata totală de execuție, de la începerea primei operații pînă la obținerea rezultatului final.

⁵⁾ ALEGE este predicatul care efectuează alegerea variantei optime din punctul de vedere al *tuturor criteriilor* considerate simultan aplicind, în cazul de față, metoda ELECTRE îmbunătățită [11, 12, 36]. Evident, poate fi folosită oricare altă metodă, cu condiția de a fi demonstrată matematic și a putea fi aplicată în condiții de cât mai puțină subiectivitate.

⁶⁾ INSCRIE este predicatul care înregistrează noua variantă aleasă, completind pentru fiecare operație condițiile care lipsesc în *SELECTAREA OPERATIEI URMATOARE*. În acest fel, varianta va fi parcursă de aici înainte *programat*, ori de câte ori condițiile vor coincide cu acelea ale variantei optime înregistrate.



Menționăm, în încheiere, că dacă *realizarea* tuturor operațiilor din programul nostru (completat cu rutina INVATARE) s-ar face automat – de exemplu, prin declansarea unor dispozitive cuplate la calculator care să fie capabile să comunice calculatorului, după executare, rezultatele, parametrii și condițiile/restricții în care au fost realizate – s-ar ajunge în cele din urmă la un *robot industrial capabil să conducă producția* pe baza unei experiențe acumulate și să se autoperfecționeze conform unei logicii proprii – aceasta din urmă putind fi și ea îmbunătățită cu timpul.

Oricum, modelul INTELEC poate fi mijlocitorul unei asemenea performanțe.



CAPITOLUL

5

Conceptul de model intelligent în conducerea operativă a producției

Cu cele de pînă acum ne-am convins că modelul INTELEC este capabil să înmagazineze informații și programe și să propună soluții de continuare a unor acțiuni în situații inedite, pe baza unor rutine de **〈SALVARE〉** și **〈INVATARE〉** și ele la rîndul lor perfectibile. La ce putem folosi toate acestea, ce înseamnă de fapt conducerea operativă a producției într-o întreprindere, în zilele noastre? O încercare de răspuns este prezentată în acest capitol. Ea se bazează pe utilizarea, în mod intelligent, a mijloacelor automate de calcul și are drept scop să ofere, în primul rînd, un cadru de aplicare în întreprinderi a unor pachete de programe pentru producție existente la noi în țară. Calculatorul-consilier în probleme de informații și proceduri de lucru dar și calculatorul-executant al unor programe complicate de optimizare, programare, nivelare a resurselor etc. corespund unor utilizări posibile la ora actuală în întreprinderi. Prelucrarea totală a unor sarcini de conducere — calculatorul-robot — deși limitată astăzi ca domenii de aplicare, apare totuși în perspectivă ca decurgînd firesc din combinarea și dezvoltarea primelor două ipostaze în care apare calculatorul (consilier și executant), la care se adaugă firesc modele adecvate ale mediului precum și modele de decizie și control al executăril deciziilor. Un exemplu concret din domeniul construcțiilor de mașini înțeinde să demonstreze, pe baza utilizării modelului INTELEC și a unor proceduri ierarhizate din ce în ce mai specifice, că un asemenea concept este viabil, asigură o dezvoltare nelimitată și corespunde necesităților stringente de adaptare a informaticii la exigențele actuale ale conducerii operative a producției.

5.1. Structura conducerii operative a producției

Conducerea operativă a producției, așa cum este definită ea de majoritatea autorilor ([14] p. 93–116; [15] p. 55–85; [46]; [55] p. 131–153; [30] p. 219–268), comportă o serie de trăsături caracteristice:

1° Este un *proces cibernetic* care se compune din comenzi (de realizare a sarcinilor de producție), urmate de execuții și apoi de controlul acestora din urmă, cărora le succed aprecierea rezultatelor și în final reluarea parțială sau integrală a procesului pe baza procedurilor cunoscute sau conform unor noi proceduri.

2° Procesul se desfășoară în limitele (cadrul) impus de o *planificare globală* a producției în care au fost stabilite: tipurile de produse și sortimentele lor împreună cu cantitățile optime de fabricat, ținându-se cont de capacitatele disponibile și de obiectivele întreprinderii la nivelul indicatorilor economici.

3° Posibilitățile de realizare ale acestor obiective se verifică în cadrul *programării de ansamblu* printr-o eşalonare în timp a produselor, în aşa fel încit suma încărcărilor să nu depășească capacitatele posturilor de lucru ale întregii producții, să aibă un grad înalt de ritmicitate și continuitate.

4° Cadrul fixat de programarea de ansamblu constituie condiții/restricții pentru *programările curente*, pe perioade mai scurte, care urmează și care țin în plus seama de condițiile locale și de fluctuațiile mediului (apariția, pe parcurs a unor contracte avantajoase la export, urgențe provocate de întirzieri sau dereglați în aprovizionare, lipsa unor înlocuitori pentru materialele din import, defectarea unor utilaje sau nefuncționarea lor la parametrii proiectați etc.)

5° Scopul întregului proces – constituit din planificarea globală, programarea de ansamblu și programările curente – este *elaborarea de sarcini pe locuri de muncă și eşalonate în timp*, în aşa fel încit, intercondiționate funcțional, să nu se contrazică reciproc și să ducă la realizarea obiectivelor de producție ale întreprinderii.

Referitor la procesele de producție discontinui, specifice construcției de mașini, o serie de specialiști și cercetători în conduceri și informatică au definit un concept de conduceri operative a producției constituit din următoarele patru faze: (i) pregătirea fabricației și programarea de ansamblu, (ii) definirea sistemului de producție și a formelor de organizare, (iii) calculul parametrilor desfășurării producției, (iv) ordonanțarea fabricației ([45] p. 14–26; [46, 48, 37]).

i. *Pregătirea fabricației* și programarea de ansamblu* are rolul de a crea condițiile pentru desfășurarea producției în anul de plan. În această fază se rezolvă următoarele sarcini: alcătuirea specificațiilor de producție (tipurile de produse pe sortimente și dimensiuni, structura și tehnologia lor de fabricație); pregătirea documentației de lansare (bonuri de lucru, bonuri de material, fișe limită, fișe de urmărire etc., pentru fiecare tip de

* Vezi „Organizarea și conducerea producției”, Academia Ștefan Gheorghiu, 1979, Secțiunea 1: Organizarea producției, p. 25–49 și Enciclopedia conducerii întreprinderii, Edit. științifică și enciclopedică, 1982.

produs) ; planul de producție ; planul de investiții ; planul de reparații și întreținere a utilajelor ; planul forței de muncă ; planul de aprovizionare ; planul de colaborări ; nivelul stocurilor.

Pe baza celor de mai sus se efectuează, mai întii, o *armonizare globală* a obiectivelor cu resursele, folosind în acest scop modelele de programare liniară pentru diferite funcții-obiectiv (minimizarea cheltuielilor materiale, maximizarea producției, maximizarea beneficiului etc.) și se obțin în final o serie de *variante de structură a planului de producție* (listă de produse și cantități optime de fabricat din punctul de vedere al obiectivelor formulate). Această operație se execută, de regulă, pe calculatorul electronic, utilizând pachete de programe specializate de tipul PROGEN sau PROGLIN (pentru calculatorul FELIX-C512) și LPS sau MPS (pentru calculatoarele IBM), capabile să rezolve în timp scurt sisteme de ecuații cu zeci de restricții și sute de necunoscute.

Odată determinate sortimentele și cantitățile de produse pe întreg anul de plan, urmează să se efectueze o *așezare a lor în timp* (programare de ansamblu) în aşa fel încât să se poată stabili (cu aproximare), ținând cont de structura ciclurilor de fabricație ale produselor și de capacitatele disponibile în fiecare lună ale posturilor de lucru :

- a) dacă se poate sau nu realiza planul propus ;
- b) dacă da, atunci :

- care sunt datele de început (intrare în fabricație) și de sfîrșit (livrare) pentru fiecare tip de produs (comandă beneficiar sau comandă internă, în funcție de modul de organizare adoptat de întreprindere respectivă),

- care sunt lunile în care se prevăd anumite dificultăți în realizarea planului (încărcări mari ale posturilor de lucru în comparație cu capacitatele de producție),

- care sunt lunile în care se prevăd unele goluri de producție pe anumite posturi de lucru (capacități de producție disponibile) ;

- c) dacă nu, atunci :

- care sunt posturile de lucru care provoacă *strangulări permanente* (supraincărcări mari pentru toate variantele de plan încercate),

- care sunt comenzi care pot fi, eventual omise, sau aminate pentru a face posibilă executarea la timp și în bune condiții a celor-lalte comenzi prioritare.

Este evident că nu toate armonizările globale, efectuate în prima parte a pregătirii, vor duce la variante realizabile din punct de vedere al programării. Dacă prin programarea de ansamblu, păstrînd restricțiile inițiale, nu se ajunge la o soluție acceptabilă pentru întreprindere, atunci este necesară adoptarea unei variante suboptimale din punctul de vedere al obiectivelor (cheltuieli materiale, beneficii etc.) dar care să poată fi efectiv realizată și acest lucru nu se poate face decât ajustînd corespunzător parametrii programării. De aceea, partea a doua a fazei de pregătire constă în *căutarea unei variante de program realizabile care să fie totodată cît mai apropiată de optimul economic al întreprinderii*.

Odată stabilită această variantă, ea va constitui documentul de bază al întreprinderii în cadrul discuțiilor de *contractare* cu beneficiarul extern și/sau intern, care vor alcătui cea de-a treia parte a fazei de pregătire. Și aici este evident că nu toate termenele, care convin întreprinderii producătoare din punctul de vedere al uniformității încărcării posturilor de lucru și care asigură o ritmicitate și continuitate înaltă a producției,

convin în egală măsură beneficiarilor externi și/sau interni. În urma discuțiilor purtate la contractare se va ajunge, de cele mai multe ori, la un compromis care să satisfacă ambele părți contractante. Important este ca această soluție de compromis să fie din nou verificată din punctul de vedere al programării, cu alte cuvinte să se aprecieze în ce măsură abaterile de la varianta realizabilă, convenabilă întreprinderii, influențează ritmicitatea și continuitatea producției, provoacă supraincărcări mari sau goluri de sarcină inadmisibile etc., și să se intervină din timp pentru eliminarea acestor deficiențe.

Dacă fiind faptul că atât partea a doua cât și partea a treia a fazei de pregătire necesită un volum enorm de calcule precum și un *algoritm de programare convergent* care să asigure obținerea în final a unei soluții satisfăcătoare, se recomandă ca ele să fie executate pe calculatorul electronic cu ajutorul unui pachet de programe de tipul SOFTPLAN*, ORDO (pe calculatorul FELIX-C512), sau CLASS (pe calculatoarele IBM).

Varianta definitivă obținută cu calculatorul după etapa de contractare va deveni *program anual de fabricație*, obligatoriu pentru întreprindere în anul de plan.

Pe baza acestui program se pot calcula, în partea a patra a fazei de pregătire :

1° Cantitățile de *materii prime, materiale și semifabricate* necesare a fi aduse din țară, de la unitățile sau bazele de aprovizionare precum și, dacă este cazul, din import, pe semestre, trimestre și luni. Acestea constituie elementele fundamentale ale *contractelor de aprovizionare* care trebuie încheiate cu furnizorii, cu mult înainte de începerea anului de plan.

2° *Lista subansamblelor* ce trebuie aduse din *colaborări* precum și datele la care este necesar ca acestea să se afle în magazia întreprinderii pentru a putea fi respectate termenile de livrare ale produselor. În continuare urmează să se ia măsuri în vederea perfectării *contractelor de colaborare* cu întreprinderile producătoare de aceste subansamblle.

3° *Lista reperelor (pieselor), subsansamblelor, ansamblelor și produselor* de fabricat pe semestre, trimestre și luni.

ACESTE liste servesc la calculul nivelului stocurilor maxime, dimensionarea și pregătirea magaziilor de depozitare, organizarea transportului intern precum și a livrărilor către beneficiar, pregătirea documentației pentru lansare în fabricație etc.

Alcătuirea tuturor listelor menționate — de materiale, piese, colaborări etc. — este automatizată prin punerea la dispoziția întreprinderilor a unor pachete de programe eficiente, larg răspândite astăzi în lume și utilizate cu succes la noi în țară, cum ar fi de exemplu : LANSARE (pe calculatorul FELIX-C512), BOMP, DIBOMP și REQUIREMENTS PLANNING (pe calculatoarele IBM). Ele se bazează pe construirea și exploatarea unor structuri arborescente de componență a produselor pînă la nivel de material.

ii. *Definirea sistemului de producție și a formelor de organizare* constă în alegerea tipului de producție (unicate, serie mică, mijlocie, mare sau producție de masă), precum și a formelor concrete de organizare a producției (serie, paralel sau serie-paralel) în funcție de condițiile locale, obiectivele întreprinderii și ținind cont de legile și principiile organizării proceselor de producție în spațiu și timp ([45] p. 14—24 ; [47]).

* Vézi capitolul 6.



În legătură cu această fază trebuie menționat faptul că, datorită ritmului înalt de dezvoltare economică înregistrat în țara noastră în ultimele două decenii — ritm și mai avansat încă în ramura construcțiilor de mașini — la multe întreprinderi și mai ales la unele produse din cadrul aceleiași întreprinderi, se înregistrează treceri frecvente de la producția de unice și serie mică la producția de serie mare și uneori de masă. De asemenea, datorită modernizării rapide a tehnologiei, unele produse se reduc mult ca pondere sau se scot chiar complet din fabricație. Această dinamică, deosebit de accentuată în actuala etapă de dezvoltare, ne face să subliniem importanța redefinirii sistemului de producție pentru fiecare tip de produs în parte precum și studierea formelor sale specifice de organizare la toate întreprinderile constructoare de mașini, la începutul fie căruia de plan.

Deși urmează logic primei faze, activitățile care țin de definirea sistemului de producție și a formelor de organizare pot fi considerate tot de *pregătire*, ele efectuindu-se, de regulă, înaintea începerii anului de plan.

Deoarece cea mai mare parte a acestor activități se bazează pe observații și calcule, ele se pot automatiza și efectua cu ajutorul calculatorului electronic.

iii. *Calculul parametrilor desfășurării producției* este tot o fază pregătitoare de calcul, așa cum reiese și din denumire, care constă în determinarea unor parametri absolut necesari desfășurării producției, cum ar fi, de exemplu, pentru producția de serie mărimea lotului optim, lungimea ciclului de fabricație, perioada de repetabilitate a ciclurilor, valoarea producției neteterminate medii ciclice, iar pentru producția de serie mare și masă numărul de mașini și forța de muncă necesară efectuării fie cărei operații. Se înțelege că și aceste calcule pot fi automatizate și efectuate pe calculator cu ajutorul unor programe adecvate. Automatizarea lor este cu atit mai necesară cu cit este știut faptul că pe parcursul desfășurării producției au loc tot felul de deregări care fac să se modifice parametri inițiali luați în calcul. Deoarece calculele în sine sunt destul de numeroase, executarea lor manuală ar fi imposibil de realizat la fiecare modificare, ceea ce ar duce la lipsa unor informații esențiale despre producție și în consecință la erori de raportare și programare.

iv. *Ordonanțarea fabricației* este faza cea mai laborioasă a conducerii operative a producției și constă în *eşalonarea în timp a sarcinilor de producție pe fiecare loc de muncă, în raport cu mărimea parametrilor producției, în aşa fel încât sarcinile să concorde în permanență cu disponibilitatea fie cărei resurse.*

Ordonanțarea este o activitate complexă care presupune, în primul rînd, adoptarea unei *metode* de ordonanțare — manuală sau automată — potrivit cu particularitățile procesului de producție și posibilitățile economice ale întreprinderii. Metodele automate presupun aplicarea pe calculator a unor algoritmi euristică de simulare a producției [65] sau a unor pachete de programe de firmă — *ORDO*, *CLASS* — precum și a unor pachete de programe mai generale dar mai bine adaptate specificului producției din țara noastră, precum *SOFTPLAN*^{*)} ([37] p. 49—53).

^{*)} Cu privire la avantajele utilizării pachetului de programe *SOFTPLAN* în comparație cu pachetele de firmă vezi [37].

Trecerea la aplicarea metodei de ordonanțare aleasă, presupune organizarea corespunzătoare a fluxului informațional. Se stabilesc și se pun la punct următoarele operații :

- conținutul informațional și forma documentelor de control a producției (bonuri de lucru, bonuri de material, fișe limită, note de comandă, note de rebut etc.). Dacă se efectuează un control riguros, cu ajutorul unui echipament automat de culegere a datelor, aceste documente își modifică forma clasică adaptindu-se echipamentelor și procedurilor de culegere utilizate ;
- procedurile de intrare și scoatere din magazii a materialelor, semifabricatelor și produselor ;
- procedurile de urmărire a executării fiecărei operații (sau a unui grup de operații) ;
- procedurile de raportare și centralizare a datelor referitoare la executarea produselor ;
- reprogramarea operațiilor întirziate ;
- luarea în considerare, la reprogramare, a unor comenzi noi, devenite prioritare pe parcursul desfășurării planului.

Ordonanțarea este o activitate care se repetă la intervale de timp mult mai mici în comparație cu celelalte faze enumerate mai sus, cu alte cuvinte este mai apropiată de înțelesul cuvântului „operativ”. Aceasta a făcut ca mulți autori să o confundă cu conducerea operativă propriu-zisă sau să adopte pentru ea denumirea de *programare operativă*.

În ceea ce ne privește, suntem pentru abordarea unitară a procesului de conducere a producției, sub toate aspectele menționate, calitatea de „operativ” având pentru noi sensul de *procedură de acțiune menită să conducă la realizarea unui proces cibernetic compus din : comandă, execuție, controlul execuției și decizie privind continuarea procesului*.



5.2. Conceptul de model intelligent

Complexitatea deciziilor ce țin de conducerea operativă a producției face imposibilă anticiparea tuturor comenziilor și a reacțiilor sistemului, în consecință, rămân zone întregi în care comportamentul acestuia din urmă este imprevizibil. Adaptarea sistemului de conducere operativă a producției la situațiile neprevăzute ar apărtine, în cazul în care conducerea s-ar efectua automat, domeniului *inteligentei artificiale*.

După părerea noastră, un model de acțiune bazat pe analiza obiectivelor întreprinderii, a parametrilor procesului de producție, a modului de îndeplinire a sarcinilor și care ar avea în vedere utilizarea anumitor instrumente de calcul — pachete de programe, limbaje specializate sau chiar automate conduse prin calculator — ar putea suplini, cel puțin în parte, lipsa de experiență, hazardul, intuiția, care însoțesc cele mai multe decizii în acest domeniu. Într-un asemenea model, rolul calculatorului electronic nu s-ar reduce la unele intervenții sporadice dar nici n-ar putea fi extins — decit în cazuri excepționale — la dimensiunea unui automat total. Poziția calculatorului, vis-à-vis de aceea a decidentului uman și

în scopul deservirii acestuia din urmă, poate îmbrăcea următoarele aspecte :

a. *Calculatorul-consilier* în problemele generale de conducere operațivă a producției, în sensul furnizării de informații, la cerere, pentru cele mai diferite situații ce pot apărea în practică. Informațiile sunt selectate pentru diferite nivale de conducere și pot fi împărțite în două mari categorii : *definiții și proceduri*, primele servind la explicarea unor noțiuni iar ultimele la indicații de acțiune în anumite cazuri concrete. Procedurile pot fi și ele de două feluri : *manuale și automate*. *Proporția între procedurile manuale și cele automate la un anumit moment istoric nu joacă un rol fundamental în definirea conceptului*.

Trecerea unor acțiuni pe seama calculatorului este un proces continuu, care îmbrăcă numeroase aspecte practice, este legat de dezvoltarea științelor și are implicații asupra societății dar *evoluează sigur în direcția creșterii ponderii acțiunilor automatizate, conduse prin calculator, în totalul operațiilor necesare pentru realizarea producției*.

b. *Calculatorul-executant* al unor proceduri complet automatizate, declanșate la cererea decidentului uman, sau chiar fără comanda expresă a acestuia din urmă, atunci cînd procedurile în cauză intervin în simularea unor situații efectuate în vederea aflării soluției celei mai avantajoase a problemei de producție formulate de decident (vezi cap. 4, rutina *DECIZIE*).

c. *Calculatorul-robot* ([24] p. 192–194), în sensul asumării simultane atât a rolului de decident cât și a aceluia de executant, și ducerea la bun sfîrșit a unor sarcini complexe, prelucrînd permanent informațiile provenite din mediul înconjurător și acționînd totodată, direct sau indirect asupra acestuia.

După cum se observă, factorul uman nu este complet eliminat în nici una din ipostazele de mai sus ([24] p. 193). Numai gradul său de participare este diferit în cele trei cazuri. În a utilizatorul pune întrebări și primește răspunsuri, mai mult sau mai puțin convenabile, care îl apropiie treptat de informația care îl interesează. În b același utilizator declanșează un lanț de operații automate care se termină cu un rezultat supus analizei și deciziei umane (sau automate) după care procesul continuă cu alți parametri, în alte condiții/restricții sau se trece la folosirea unei alte proceduri existente. Pentru aplicarea acesteia din urmă se pot cere noi informații de la calculator, revenindu-se astfel la primul caz ș.a.m.d. Modul de utilizare al calculatorului în cazurile a și b poartă numele de *procedeu interactiv* fiind de fapt o combinație a puterii de calcul a mașinii cu experiența și creativitatea gîndirii umane. *Datorită supleței și economicității acestei simbioze, procedeele interactive sănătoase, după părerea noastră, destinate să acopere cea mai mare parte a necesităților actuale în domeniul conducerii operative a producției*. În sfîrșit, cazul c se întâlnește mai ales în conducerea proceselor de producție cu flux continuu (rafinării, combinate chimice, combinate metalurgice și siderurgice etc.) și constituie utilizarea cea mai completă a calculatorului. Rolul omului se reduce aici la supravegherea procesului și intervenția în caz de eroare sau defectiune a robotului.

Modelul capabil să evolueze în cele trei direcții a, b și c semnalate de noi mai sus, de la simplu la complex, acoperind o gamă din ce în ce mai mare de operații și preluînd funcțiuni ale gîndirii și activității umane, va

fi denumit, în cele ce urmează, model de acțiune intelligent sau, pe scurt, model intelligent (M.I.) în conducerea operativă a producției.

5.3. Aplicarea conceptului de model intelligent în conducerea operativă a producției

În cele ce urmează ne vom strădui să demonstrăm utilitatea conceptului de model intelligent pentru conducerea operativă a producției, referindu-ne în special la procesele de producție din ramura construcțiilor de mașini. Ca exemplu, vom relua conceptul de conducere operativă a producției expus în prima parte a capitolului servindu-ne de modelul INTELEC (capitolul 3) și, pe baza principiilor de planificare a comportamentului (capitolul 4), vom încerca să-l transpunem într-o serie de proceduri, manuale și automate, supravegheate de calculator. În acest fel, participarea calculatorului la conducerea operativă a producției va fi de tipul $a + b$ de mai sus, existind, bineînțeles, tendința ca, de la caz la caz și în funcție de progresul tehnic, unele zone să devină treptat complet automatizate (de exemplu ordonanțarea și controlul fabricației).

Să ne situăm așa dar, pe poziția unei *întreprinderi constructoare de mașini*, având un sistem de producție de tip serie mică și unică, care se află în faza de *pregătire* a începerii producției (vezi i). Ce procedură generală de supraveghere pe calculator a acestei faze am putea construi în acest caz?

Vom lua în considerare ordinea firească a acțiunilor care trebuie întreprinse: (1) alcătuirea specificațiilor de producție și (2) pregătirea documentației de lansare, urmate de un *proces iterativ* care are drept scop transformarea ansamblului de indicatori economici planificați ai întreprinderii (producția netă, producția marfă, cheltuielile la 1 000 lei fonduri fixe etc.) într-un plan fizic, adică într-o listă de produse ce trebuie executate în anumite cantități, eventual și cu anumite restricții de termene, în anul de plan, toate acestea ținând seama de resursele reale ale întreprinderii (capacității de producție, forță de muncă, plan de investiții, colaborări etc.). Se folosesc în cadrul acestui proces iterativ, așa cum am arătat la începutul acestui capitol, modele de programare liniară cu una sau mai multe funcții-obiectiv și pachete de programe pe calculatorul electronic. Rezultatele armonizărilor globale între obiective și resurse se verifică și se ajustează printr-o programare de ansamblu pe calculatorul electronic, folosind, de exemplu, pachetul de programe SOFTPLAN. Abia în urma acestor programări planul fizic este înaintat spre aprobare forurilor superioare (centrală, minister) și se întoarce în întreprindere ca plan de stat, obligatoriu pentru anul viitor.

Următoarea procedură, în limbaj INTELEC, ar putea servi ca model în această etapă:

⟨PREGATIRE⟩

1. ⟨SE⟩ [ALOCUIESTE] ⟨SPECIFICATIE DE PRODUCȚIE⟩.
2. ⟨SE⟩ [PREGATESTE] ⟨DOCUMENTATIE DE LANSARE⟩.
3. ⟨SE⟩ [DEFALCA] (PL) ⟨INDICATOR⟩ (PE) (PL) ⟨INTREPRINDERE⟩.

4. <SE> [TRANSFORMA] (PL) <INDICATOR> (AL/A) <INTreprindere> (IN) <PLAN FIZIC> (AL/A) <INTreprindere>.
5. <SE> [VERIFICA REALIZAREA] <PLAN FIZIC> (AL/A) <INTreprindere>.
6. <DACA> <REZULTAT> [ESTE NESATISFACATOR] <ATUNCI> <SE> [MERGE LA]* (3) <FRAZA>*.
7. <SE> [ADOPTA] <VARIANTA OPTIMA> (AL/A) <PLAN FIZIC> (AL/A) <INTreprindere>.
8. <VARIANTA OPTIMA> (AL/A) <PLAN FIZIC> (AL/A) <INTreprindere> [ESTE (IN ACEST CAZ)] <PROGRAM ANUAL>

în care :

- a. <INDICATOR> se referă la indicatorii obișnuiți de planificare la nivelul de decizie al eșalonului superior.
- b. [DEFALCA] indică o operație, manuală sau automată, de defalcare a indicatorilor pe fiecare întreprindere.
- c. [TRANSFORMA] cuprinde alcătuirea liste de produse, a cantităților și, eventual, a restricțiilor de termene pentru fiecare tip de produs.
- d. [VERIFICA REALIZAREA] conține simularea producției și programarea în timp a fabricației fiecărui tip de produs, terminată cu obținerea variantelor care conduc la o soluție satisfăcătoare. Acest lucru se poate face, de exemplu, cu ajutorul pachetului de programe SOFTPLAN prin <PROCEDURA DE UTILIZARE SOFTPLAN> (vezi capitolul 3) în care <PROCEDURA DE OPTIMIZARE SOFTPLAN> se execută conform <PROGRAM PRINCIPAL> (vezi capitolul 4). Dacă rezultă imposibilitatea respectării restricțiilor impuse, se prezintă conducerii listele care permit analizarea cauzelor pentru care nu se poate ajunge la o programare satisfăcătoare (lipsă de capacitate pe anumite posturi de lucru, cantități prea mari de produse, restricții prea mari în privința termenelor de livrare etc.).
- e. Frazele 6, 7 și 8 exprimă decizia eșalonului superior privind acceptarea sau respingerea variantei optime prezentate de întreprindere. Dacă neconcordanțe cu alte întreprinderi, perturbații în dezvoltarea ramurii etc. se opun adoptării soluției, atunci se dau indicații pentru modificarea unor parametri sau condiții/restricții și problema se reia de la fraza 3. Odată adoptată, varianta optimă devine lege și va servi ca bază programărilor ulterioare, pe perioade mai scurte, în faza de ordonanțare.

Tinind seama de cele menționate în cadrul capitolului 3 cu privire la generalizarea și particularizarea noțiunilor, baza de date relațională trebuie completată cu informații suplimentare, de tipul :

<INDICATOR> [SE COMPUNE DIN (ANSAMBLU)] <PRODUCTIE NETA> (SI) <PRODUCTIE MARFA> (SI) <CHELTUIELI LA 1000 LEI FONDURI FIXE> (SI) ... <INTreprindere> [ESTE (IN ACEST CAZ)] <INTreprinderea X> <PLAN FIZIC> [SE COMPUNE DIN (ANSAMBLU)] (PL) <PRODUS> <PRODUS> [SE COMPUNE DIN (ANSAMBLU)] <COD PRODUS> (SI) <CANTITATE> (SI) <PRIORITATE> (SI) <TERMEN DE INCEPUT CEL MAI DEVREME> (SI) <TERMEN DE SFIRSIȚ CEL MAI TIRZIU> <REZULTAT> [SE COMPUNE DIN (ANSAMBLU)] <NIVELUL SUPRAINCARCARILOR RESTANTE> (SI) <NIVELUL GOLURILOR DE PRODUCTIE> (SI) <INVESTITII SUPLIMENTARE> (SI) <COLABORARI> (SI) <ORE SUPLIMENTARE> (SI) <DECALARI> (SI) <RENUNTARI> etc.

Noțiunile <PRODUS> și <REZULTAT> sunt detaliate conform cerințelor de aplicare, în acest caz, a pachetului de programe SOFTPLAN (vezi capitolul 4). Astfel, <PRODUS> conține o serie de informații (cod, cantitate, prioritate etc.) care urmează să fie particularizate pentru fiecare comandă



în parte. De asemenea, **〈RESULTAT〉** se referă la unele aprecieri cantitative și calitative ale rezultatelor obținute cu acest pachet de programe. Analizând fiecare rezultat final, utilizatorul va fi obligat să aprecieze: nivelul supraîncărcărilor, nivelul golurilor de producție, investițiile suplimentare necesare eliminării locurilor inguste, colaborările propuse pentru rezolvarea cazurilor limită, numărul de ore suplimentare, decalările de termene sau comenziile la care ar trebui să se renunțe, pentru fiecare variantă prezentată conducerii în vederea adoptării planului anual de producție al întreprinderii.

Faza de pregătire se încheie, așa cum rezultă din descrierea efectuată la începutul acestui capitol, cu calculul necesarului de materii prime, materiale și semifabricate de aprovizionat, a necesarului de piese, subansamblu și produse de fabricat și alcătuirea listei de colaborări pe subperioade (semestre, trimestre, luni). Pentru a trece și această activitate sub controlul calculatorului este necesară completarea rutinei **〈PREGATIRE〉** de mai sus cu următoarele fraze:

〈PREGATIRE〉

9. **〈SE〉 [IMPARTE] 〈PROGRAM ANUAL〉 (IN) (PL) 〈SUBPERIOADA〉**
10. **〈SE〉 [ALCATUIESTE] 〈LISTA PRODUSE DE FABRICAT〉 (PENTRU) (FIECARE) 〈SUBPERIOADA〉**
11. **〈SE〉 [CALCULEAZA NECESAR MATERIALE] (PENTRU) (FIECARE) 〈LISTA PRODUSE DE FABRICAT〉**
12. **〈SE〉 [CALCULEAZA NECESAR COLABORARI] (PENTRU) (FIECARE) 〈LISTA PRODUSE DE FABRICAT〉**
13. **〈SE〉 [CALCULEAZA NECESAR PIESE, ANSAMBLE] (PENTRU) (FIECARE) 〈LISTA PRODUSE DE FABRICAT〉**

pentru care, am dat în continuare următoarele explicații:

f. **[IMPASTE]** indică operațiunea de divizare, efectuată manual de utilizator, a anului de plan în semestre, trimestre, luni, acestea din urmă simbolizate generic prin noțiunea **〈SUBPERIOADA〉**.

g. **[ALCATUIESTE]** se referă la extragerea, pentru fiecare **〈SUBPERIOADA〉** a produselor a căror fabricație ar urma să înceapă în intervalul respectiv (conform programului anual adoptat).

h. **[CALCULEAZA NECESAR MATERIALE]** cuprinde calculul, efectuat de astă dată automat, cu unul din pachetele de programe amintite, a materiilor prime, materialelor, semifabricatelor etc. care rezultă din descompunerea produselor pentru fiecare **〈SUBPERIOADA〉**. Dacă se mai dorește, de exemplu, și o listă cu materialele necesare din import, atunci se adaugă fraza :

- 11'. **〈SE〉 [CALCULEAZA NECESAR MATERIALE DIN IMPORT] (PENTRU) (FIECARE) 〈LISTA PRODUSE DE FABRICAT〉**

în care predicatul are semnificația specială menționată.

i. **[CALCULEAZA NECESAR COLABORARI]** se referă la semifabricatele necesare a fi aduse de la alte întreprinderi în perioada respectivă **〈SUBPERIOADA〉** pentru a se putea garanta îndeplinirea planului. Rezultatul se obține prin exploatarea pachetelor de programe de calcul al necesarului.

j. [CALCULEAZĂ NECESSAR PIESE, ANSAMBLE] indică un calcul detaliat, efectuat cu aceleasi pachete de programe, pentru tot ceea ce trebuie fabricat in intreprindere in 〈SUBPERIOADA〉 respectivă.

★

Fazele *ii* și *iii* ale conducerii operative a producției sint, așa cum am văzut, faze de calcul. Înserierea lor in baza de date relațională se face utilizind predicatul intern [EXECUTA] * alături de un complement specific în stare să declanșeze o serie de calcule executate manual sau, dacă se dispune de programe, cu ajutorul calculatorului electronic. Avem deci următoarele rutine de calcul :

〈DEFINIREA SISTEMULUI DE PRODUTIE SI A FORMELOR DE ORGANIZARE〉

〈SE〉 [EXECUTĂ]* 〈CALCULUL SISTEMULUI DE PRODUCȚIE〉
 〈SE〉 [EXECUTĂ]* 〈ALEGEREA FORMEI DE ORGANIZARE (PENTRU)
 (FIECARE) (REPER I)〉

și :

〈CALCULUL PARAMETRILOR DESFASURARII PRODUCȚIEI〉

〈SE〉 [EXECUTĂ]* 〈CALCULUL PARAMETRILOR〉 (PENTRU) (FIECARE)
 (REPER I).

Utilizind metoda indicilor de constanță a fabricației pentru stabilirea sistemului de producție in construcția de mașini ([45] p. 14—24) se poate folosi, de exemplu, algoritmul de mai jos pentru automatizarea operațiilor pe calculatorul electronic :

||Calculează fondul nominal de timp|| $F \square$
 pentru toate ||reperele (a_i) $i = 1, \dots, m$ din 'LISTA REPERELOR'||

||Citește datele pentru reperul a_i || \square

||Calculează volumul anual de piese N_{a_i} || \square

||Calculează ritmul mediu anual de fabricație ρ_{a_i} || \square

$N_a, N_a^1, N_a^2, N_a^3, N_a^4 \leftarrow 0 \square$

pentru toate ||operațiile (o_{ij}) , $j = 1, \dots, n$ din ('LISTA OPERAȚIILOR') $_i$ ||
 ||Citește datele referitoare la operația j || \square

$N_a \leftarrow N_a + 1 \square$

||Calculează consumul de timp pe o operație t_{ij} || \square

||Calculează coeficientul sistemului de producție K_{ij} || \square

dacă $K_{ij} \leq 1$ atunci

$N_a^1 \leftarrow N_a^1 + 1$

||Listează 'o_{ij} = PRODUCȚIE DE MASA'||
 \square

dacă $1 < K_{ij} \leq 10$ atunci

$N_a^2 \leftarrow N_a^2 + 1$

||Listează 'o_{ij} = PRODUCȚIE DE SERIE MARE'〉

□ dacă $10 < K_{ij} \leq 20$ atunci

$$N_a^3 \leftarrow N_a^3 + 1$$

||Listează ' o_{ij} = PRODUCȚIE DE SERIE MIJLOCIE'||

□ $N_a^4 \leftarrow N_a^4 + 1$

||Listează ' o_{ij} = PRODUCȚIE DE SERIE MICA'||

□ ||Calculează $\max_{l=1 \dots 4} (N_a^l / N_a) = N_a^{l^*} / N_a||$

dacă $l^* = 1$ atunci

'SISTEMUL DE PRODUCȚIE PREDOMINANT PENTRU REPERUL a_i ESTE PRODUCȚIA DE MASA'

□ dacă $l^* = 2$ atunci

'SISTEMUL DE PRODUCȚIE PREDOMINANT PENTRU REPERUL a_i ESTE PRODUCȚIA DE SERIE MARE'

□ dacă $l^* = 3$ atunci

'SISTEMUL DE PRODUCȚIE PREDOMINANT PENTRU REPERUL a_i ESTE PRODUCȚIA DE SERIE MIJLOCIE'

□ 'SISTEMUL DE PRODUCȚIE PREDOMINANT PENTRU REPERUL a ESTE PRODUCȚIA DE SERIE MICA'

in care fondul nominal de timp F se calculează cu :

$$F = Z_i \cdot K_s \cdot h \text{ (ore/an)} \quad (5.1)$$

unde Z_i este număr de zile lucrătoare anual, K_s – număr de schimburi lucrătoare pe zi, h – număr de ore lucrătoare pe schimb.

Volumul anual de piese rezultă aplicând formula :

$$N_{ai} = \sum_{j=1}^w (q_{ij} \cdot N_j + N'_j + N''_j) (1 + \beta/100), \quad (5.2)$$

unde q_{ij} este numărul de piese de tip a_i în produsul j ; N_j – volumul anual de produse j ; N'_j – volumul de piese de schimb de tip a_i , necesare pentru sortimentul j ; N''_j – stocul de producție neterminată, de siguranță, de piese a_i pentru sortimentul j ; β – procentul admisibil de rebut tehnologic.

Ritmul mediu anual de fabricație este dat de formula

$$\rho_{ai} = (F/N_{ai}) \cdot 60 \text{ (min./buc.)}. \quad (5.3)$$

Am considerat că există m repere în 'LISTA REPERELOR' și pentru fiecare din ele se execută n operații tehnologice, durata fiecărei operații fiind dată de următorul raport

$$t_{ij} = \bar{t}_{ij}/\bar{k}, \quad (5.4)$$

unde t_{ij} este timpul normat pentru prelucrarea piesei a_i la operația j (in min/buc.), k – coeficientul mediu de indeplinire a normelor de timp pe uzină, iar coeficientul sistemului de producție este calculat cu formula

$$k_{ij} = \rho_{ai}/t_{ij}, \quad (5.5)$$

unde ρ_{ai} și t_{ij} au semnificațiile de mai sus.

În limbaj INTELEC, rutina de mai sus ar arăta astfel :

⟨CALCULUL SISTEMULUI DE PROducțIE⟩

⟨SE⟩ [EXECUTA] * ⟨CALCULUL FONDULUI NOMINAL DE TIMP F⟩
 ⟨SE⟩ [EXECUTA] * ⟨CALCULUL COEFICIENTULUI SISTEMULUI DE PROducțIE⟩ (PENTRU) (FIECARE) (REPER I).

La rîndul său, calculul coeficientului sistemului de producție ar conține următoarele fraze :

⟨CALCULUL COEFICIENTULUI SISTEMULUI DE PROducțIE⟩

⟨SE⟩ [EXECUTA]* ⟨CALCULUL VOLUMULUI ANUAL DE PIESE N.I⟩
 ⟨SE⟩ [EXECUTA]* ⟨CALCULUL RITMULUI MEDIU ANUAL DE FABRICATIE RO.I⟩
 ⟨SE⟩ [ACTUALIZEAZA]* ⟨0⟩ (IN) ⟨NA⟩ (SI) ⟨NA1⟩ (SI) ⟨NA2⟩ (SI) ⟨NA3⟩ (SI) ⟨NA4⟩
 ⟨SE⟩ [EXECUTA] * ⟨CALCULUL COEFICIENTULUI K.I.J.⟩ (PENTRU) (FIECARE)
 ⟨OPERATIE O.I.J.⟩ (AL/A) (REPER I)
 ⟨SE⟩ [EXECUTA]* ⟨CALCULUL MAX. L (NA.L/NA) L = 1,4⟩
 ⟨DACA⟩ ⟨L⟩ [ESTE EGAL CU]* ⟨1⟩ ⟨ATUNCI⟩ [LISTEAZA]* ⟨REPER I⟩ (SI)
 ⟨'PROducțIE DE MASA'⟩
 ⟨DACA⟩ ⟨L⟩ [ESTE EGAL CU]* ⟨2⟩ ⟨ATUNCI⟩ [LISTEAZA]* ⟨REPER I⟩ (SI)
 ⟨'PROducțIE DE SERIE MARE'⟩
 ⟨DACA⟩ ⟨L⟩ [ESTE EGAL CU]* ⟨3⟩ ⟨ATUNCI⟩ [LISTEAZA]* ⟨REPER I⟩ (SI)
 ⟨'PROducțIE DE SERIE MIJLOCIE'⟩
 ⟨SE⟩ [LISTEAZA]* ⟨REPER I⟩ (SI) ⟨'PROducțIE DE SERIE MICA'⟩.

În continuare, calculul coeficientului K_{ij} se detaliază astfel :

⟨CALCULUL COEFICIENTULUI K.I.J.⟩

⟨SE⟩ [ADUNA]* ⟨1⟩ (LA) ⟨NA⟩
 ⟨SE⟩ [EXECUTA]* ⟨CALCULUL CONSUMULUI DE TIMP PE OPERATIE T.I.J.⟩
 ⟨SE⟩ [EXECUTA]* ⟨CALCULUL K.I.J.⟩
 ⟨DACA⟩ ⟨K.I.J.⟩ [ESTE MAI MIC SAU EGAL OU]* ⟨1⟩
 ⟨ATUNCI⟩ ⟨SE⟩ [ADUNA]* ⟨1⟩ (LA) ⟨NA1⟩ (SI)
 ⟨SE⟩ [LISTEAZĂ]* ⟨OPERATIE I, J.⟩ (SI) ⟨'PROducțIE DE MASA'⟩
 ⟨DACA⟩ ⟨K.I.J.⟩ [ESTE MAI MARE DECIT]* ⟨1⟩ (SI) ⟨K.I.J.⟩
 [ESTE MAI MIC SAU EGAL CU]* ⟨10⟩
 ⟨ATUNCI⟩ ⟨SE⟩ [ADUNA]* ⟨1⟩ (LA) ⟨NA2⟩ (SI)
 ⟨SE⟩ [LISTEAZĂ]* ⟨OPERATIE I, J.⟩ (SI) ⟨'PROducțIE DE SERIE MARE'⟩
 ⟨DACA⟩ ⟨K.I.J.⟩ [ESTE MAI MARE DECIT]* ⟨10⟩ (SI) ⟨K.I.J.⟩ [ESTE MAI MIC
 SAU EGAL CU]* ⟨20⟩

<ATUNCI> <SE> [ADUNA]* <1> (LA) <NA3> (SI)
 <SE> [LISTEAZA]* <OPERATIE I, J> (SI) <'PRODUCTIE DE SERIE MILIO-
 CIE'>
 <SE> [ADUNA]* <1> (LA) <NA4> (SI) <SE> [LISTEAZA]* <OPERATIE I, J> (SI)
 <'PRODUCTIE DE SERIE MICA'>
 s.a.m.d.

Determinarea formelor de organizare este o operație care se face, de regulă, manual, respectând indicațiile de principiu din literatura de specialitate [48].

În ceea ce privește parametrii desfășurării producției, calculul lor este, de asemenea, indicat în lucrările de specialitate [46] și se poate automatiza cu ajutorul calculatorului. Important este faptul că *toți acești parametri depind unul de altul și sunt funcție de forma de organizare aleasă: serie, paralelă sau mixtă*.

Astfel durata ciclului de fabricație T_c este funcție de mărimea n a lotului, aceasta din urmă având un optim din punct de vedere economic n_E . Dependența este liniară, de forma

$$T_c = \alpha + \beta n, \quad (5.6)$$

dar coeficienții α și β diferă pentru cele trei forme de organizare — serie, paralelă și mixtă — fiind calculați cu formule distințe. Perioada R de repetare a loturilor depinde, de asemenea, de mărimea n a lotului.

Calculul producției neterminate medii ciclice P_n se face ținând seama de *toți* cei trei parametri — n , T_c și R — de mai sus

$$P_n = n \cdot T_c / R \quad (5.7)$$

În sfîrșit, cheltuielile totale de producție pe unitatea de reper sunt și ele diferențiate în funcție de forma de organizare adoptată.

Rezultă că nu se poate lua o decizie științifică cu privire la forma de organizare necesară producției unuia sau altuia din repere, fără a cunoaște parametrii acestora din urmă, calculați în ipoteza adoptării unei forme de organizare precise. Această concluzie ne conduce la alcătuirea următoarei *scheme de interferență* între cele două faze *ii* și *iii* ale procesului de conducedere operativă a producției :

<CALCULUL PARAMETRILOR>

<SE> [CALCULEAZA]¹⁾ <LOTUL DE FABRICATIE OPTIM ECONOMIC NE.I>
 <SE> [ADOPTA]²⁾ <LOTUL DE FABRICATIE N.I.>
 <SE> [CALCULEAZA]¹⁾ <CICLUL DE FABRICATIE TC. I, K> (PENTRU) (FIECARE)
 <FORMA DE ORGANIZARE K>
 <SE> [CALCULEAZA]¹⁾ <PERIOADA DE REPETARE R.I>
 <SE> [CALCULEAZA]¹⁾ <PRODUCTIA NETERMINATA FIZICA PN. I, K>
 (PENTRU) (FIECARE) <FORMA DE ORGANIZARE K>
 <SE> [CALCULEAZA]¹⁾ <PRODUCTIA NETERMINATA VALORICA PNV. I, K>
 (PENTRU) (FIECARE) <FORMA DE ORGANIZARE K>
 <SE> [CALCULEAZA]¹⁾ <CHELTUIELILE TOTALE PE UNITATEA DE REPER C.I, K>
 (PENTRU) (FIECARE) <FORMA DE ORGANIZARE K>

⟨ALEGEREA FORMEI DE ORGANIZARE⟩

⟨SE⟩ [EXECUTA]^{*} ⟨CALCULUL PARAMETRILOR⟩ (PENTRU) ⟨REPER I⟩
⟨SE⟩ [CALCULEAZA]¹⁾ ⟨VALOAREA TOTALA A PRODUCIEI NETERMINATE MEDII
CICLICE PNT. K⟩ (PENTRU) (FIECARE) ⟨FORMA DE ORGANIZARE K⟩
⟨SE⟩ [ADOPTA]³⁾ ⟨FORMA DE ORGANIZARE⟩ (PENTRU) ⟨REPER I⟩

în care :

¹⁾ [CALCULEAZA] indică un calcul precizat mai departe prin parametrul complement. Dacă acesta din urmă are un singur indice (*i*) el este constant pentru cele trei forme de organizare dar diferit în funcție de reper. Dacă conține doi indici (*i, k*), primul (*i*) se referă la reper iar al doilea (*k*) la forma de organizare. Calculele se desfășoară în conformitate cu formulele indicate de literatura de specialitate ([46] p. 672–675).

²⁾ [ADOPTA] urmat de complementul ⟨LOTUL DE FABRICATIE N.I.⟩ se referă la alegerea mărului lotului n_i ; Aceasta din urmă este, în funcție de restricțiile locale (număr mic de produse de același tip, necesitatea eșalonării producției anumitor repere de-a lungul întregului an etc.) cît mai apropiat ca valoare de lotul optim economic n_{Ei} determinat la pasul anterior.

³⁾ [ADOPTA] urmat de complementul ⟨FORMA DE ORGANIZARE⟩ indică o acțiune manuală de hotărîre asupra formei de organizare cea mai potrivită pentru ⟨REPER I⟩, pe baza calculelor efectuate și ținând cont de eventualele restricții cu caracter local ce nu au putut fi formalizate și luate în considerare în calcul.

★

Faza *iv* a conducerii operative a producției, *ordonanțarea fabricației*, se realizează prin aplicarea unei proceduri combine, manuale și automate. Odată fixată metoda și ales algoritmul de ordonanțare, aplicarea pachetului de programe care transpune acest algoritm în operații executabile pe calculator se efectuează conform unor *indicații de utilizare*. Pentru exemplificare, dăm mai jos indicațiile generale de utilizare în cazul aplicării pachetului de programe SOFTPLAN în faza de ordonanțare a fabricației la o întreprindere cu profil de construcții de mașini.

⟨ORDONANTARE⟩

⟨SE⟩ [ALEGE] ⟨INTERVAL OPERATIV⟩¹⁾
⟨SE⟩ [ALEGE] ⟨SUBPERIOADA⟩²⁾
⟨SUBPERIOADA⟩ [ESTE (ÎN ACEST CAZ)] ⟨SUBMULTIPLU⟩ (AL/A)
⟨INTERVAL OPERATIV⟩
⟨SE⟩ [DEFALCA PE INTERVAL OPERATIV] ⟨PROGRAM ANUAL⟩³⁾ (IN)
⟨SUBPROGRAM ANUAL⟩
⟨SUBPROGRAM ANUAL⟩ [ESTE (ÎN ACEST CAZ)] (PL) ⟨COMANDA-
PRODUS DE FABRICAT⟩
⟨SE⟩ [DEFALCA PE SUBPERIOADE] ⟨CAPACITATE⟩⁴⁾ (AL/A) (PL)
⟨POST DE LUCRU⟩
⟨SE⟩ [RECALCULEAZĂ PE SUBPERIOADE] ⟨CICLOGRAMA DE FABRICATIE⟩⁵⁾
(AL/A) (PL) ⟨PRODUS⟩
⟨SE⟩ [INCARCA] (PL) ⟨COMANDA-PRODUS DE FABRICAT⟩⁶⁾ (AL/A)
⟨SUBPROGRAM ANUAL⟩

⟨SE⟩ [ADAUGA] (PL) ⟨COMANDA-PRODUS IN CURS⟩ (SI) (PL)
 ⟨COMANDA-PRODUS RESTANTA⟩ (SI) (PL) ⟨COMANDA-PRODUS NOUA,
 PRIORITARA⟩⁷⁾

⟨SE⟩ [SCOATE] (PL) ⟨COMANDA-PRODUS AMINATA⟩⁸⁾

⟨SE⟩ [EXECUTA]* ⟨PROGRAM PRINCIPAL⟩ (AVIND) ⟨PROCEDURA⟩ (EGALA CU)
 ⟨PROCEDURA DE OPTIMIZARE SOFTPLAN⟩

în care :

⁴) [ALEGE] urmat de complementul ⟨INTERVAL OPERATIV⟩ este acțiunea prin care din ⟨PROGRAM ANUAL⟩ stabilit în decursul fazei de ⟨PREGATIRE⟩ se alege un interval anume (semestrul x , trimestrul y , luna z etc.) pentru care urmează să se efectueze ordonanțarea. Acest interval (de exemplu trimestrul III al anului în curs) va purta, de aici înainte, numele de ⟨INTERVAL OPERATIV⟩.

⁵) [ALEGE] urmat de complementul ⟨SUBPERIOADA⟩ se referă la stabilirea perioadei de calcul a încărcărilor de manoperă și a disponibilului de capacitate pentru întreaga ordonanțare ce urmează să se efectueze. De fapt, așa cum se menționează în fraza următoare, ⟨SUBPERIOADA⟩ nu este altceva decit un submultiplu al ⟨INTERVAL OPERATIV⟩ ales în așa fel încit să convingă programării (exemplu : săptămîna, în cadrul unui trimestru).

⁶) [DEFALCA PE INTERVAL OPERATIV] urmat de complementul ⟨PROGRAM ANUAL⟩ este acțiunea prin care, din întregul ⟨PROGRAM ANUAL⟩ se separă numai comenziile-produs ce urmează să se execute în ⟨INTERVAL OPERATIV⟩ ales (exemplu : ⟨COMANDA X⟩, ⟨COMANDA Y⟩ și ⟨COMANDA Z⟩, care se execută în trimestrul III al anului în curs). Acestea se numesc, așa cum menționează fraza următoare, (PL) ⟨COMANDA-PRODUS DE FABRICAT⟩.

⁷) [DEFALCA PE SUBPERIOADE] urmat de complementul ⟨CAPACITATE⟩ se referă la operațiunea de recalcularare a capacitatii disponibile pe fiecare post de lucru, ținînd cont de ⟨SUBPERIOADA⟩ aleasă (exemplu : capacitatea postului W , figura cu 612 ore/lună în ⟨PROGRAM ANUAL⟩. Pentru ordonanțarea care se efectuează săptămînal, aceeași capacitate se exprimă ca 47 ore/săptămînă).

⁸) [RECALCULEAZA PE SUBPERIOADE] urmat de complementul ⟨CICLOGRAMA DE FABRICATIE⟩ are aceeași semnificație de recalcularare, de data aceasta a încărcărilor din ⟨CICLOGRAMA DE FABRICATIE⟩ a fiecărui produs. (exemplu : dacă produsul X avea pentru postul de lucru Z , în ⟨PROGRAM ANUAL⟩, încărcarea de 500 ore în prima lună relativă de la începutul ciclului, urmată de 300 ore în cea de-a doua lună, pentru ⟨ORDONANTARE⟩ în care am ales ca ⟨SUBINTERVAL⟩ săptămîna, aceste cifre trebuie exprimate în încărcări săptămînale. În consecință, vom obține pentru exemplul de față, prin împărțirea la 4, o ciclogramă compusă din : 125 ore în prima săptămînă relativă, 125 ore în cea de-a doua săptămînă relativă și.a.m.d., iar începînd cu a 5-a săptămînă relativă 75 ore/săptămînă pînă la sfîrșitul ciclului).

⁹) [INCARCA], urmat de complementul ⟨COMANDA-PRODUS DE FABRICAT⟩ indică operațiunea de încărcare în baza de date a pachetului de programe SOFTPLAN, a comenziilor-produs defalcate din ⟨PROGRAM ANUAL⟩ pentru ⟨INTERVAL OPERATIV⟩ considerat ; Aceasta este așa-numita *încărcare de bază*, în care intră, în mod obișnuit, majoritatea comenziilor.

⁹) [ADAUGA], urmat de o serie de complemente care arată ce anume trebuie adăugat, este un predicat care indică o eventuală *suplimentare a*

încărcării bazei de date SOFTPLAN în vederea ordonanțării. În cazul de față este vorba de :

a) Adăugarea eventualelor comenzi-produs aflate în curs de fabricație, (PL) **«COMANDA-PRODUS ÎN CURS»**, evident, numai a părții restante în **«INTERVAL OPERATIV»** avut în vedere ; De regulă, aceste comenzi primesc prioritate zero și rămîn fixate la începutul intervalului ^{*)}.

b) Adăugarea eventualelor comenzi-produs restante din **«INTERVAL OPERATIV»** precedent, care nu au putut fi începute pînă acum din diverse motive.

c) Adăugarea eventualelor comenzi-produs noi, în general prioritare față de restul comenziilor, care nu au fost prevăzute în **«PROGRAM ANUAL»** dar care apar datorită condițiilor locale avantajoase (export, necesitatea dării mai devreme în exploatare a unor obiective de investiții etc.).

^{*)} [SCOATE], urmat de o serie de complemente care arată ce anume trebuie scos, este un predicat care indică posibila *reducere* a **«PROGRAM ANUAL»** prevăzut inițial, datorită modificării unor condiții pe parcursul anului de plan. În general, reducerile (sau amînările pentru următorul **«INTERVAL OPERATIV»**) sunt efectuate tocmai pentru a face loc noilor comenzi-produs prioritare menționate la punctul c) anterior.

^{*)} **«PROGRAM PRINCIPAL»** este de fapt programul de realizare a unei **«PROCEDURA»** (în cazul nostru **«PROCEDURA DE UTILIZARE SOFTPLAN»**) în care operațiile pot fi manuale sau automate și în care pregătirea, execuția și înregistrarea rezultatelor se face apelind un **«SUBPROGRAM»**. Structura acestora a fost dată în cadrul capitolului 4 atunci cînd s-a discutat despre planificarea comportamentului cu ajutorul modelului INTELEC. Deoarece exemplul avut în vedere acolo era tocmai acela al utilizării în întreprinderi a pachetului de programe SOFTPLAN, nu mai repetăm și aici această procedură. Menționăm numai că toate considerațiile și concluziile elaborate a acel capitol rămîn valabile și în cazul de față, atunci cînd ne referim a ordonanțare.

^{*)} Vezi capitolul 6.



CAPITOLUL

6

Sistemul SOFTPLAN –element de bază în realizarea unui model intelligent pentru conducerea operativă a producției în ramura construcțiilor de mașini

Ce este sistemul SOFTPLAN? Ce calități ale acestui sistem ne determină să-l considerăm element de bază în realizarea unui model intelligent pentru construcția de mașini, de tipul celui prezentat în cadrul capitolului precedent? În cele ce urmează este descris în detaliu acest sistem original, mai puțin cunoscut la ora actuală, în comparație cu pachetele de programe de firmă, consacrate deja în literatura de specialitate. Modelul matematic, arhitectura sistemului, schemele logice generale și concepția bazei de date, constituie cîteva din punctele de atracție ale acestui capitol. În încheiere sunt menționate cele mai reprezentative rezultate obținute pînă în prezent cu acest sistem la unele întreprinderi constructoare de mașini și metalurgice din țară. Ținînd seama de experiența de pînă acum în ceea ce privește aplicarea concretă a sistemului SOFTPLAN la diverse întreprinderi se propun unele acțiuni menite să ducă la creșterea eficacității de utilizare precum și la ușurința rezolvării problemelor de adaptare, prin integrarea sa într-un model intelligent de conducere operativă a producției de tip INTELEC.

6.1. Utilitatea sistemului

În ultima parte a capitolului anterior, cu ocazia prezentării conceputului de model intelligent în ramura construcțiilor de mașini, s-a subliniat rolul deosebit de important pe care-l deține în cadrul acestui model sistemul SOFTPLAN. Pachetul de programe SOFTPLAN poate fi utilizat ca instrument științific în ajutorul decidențului, atât în faza de planificare — pentru verificarea posibilităților de realizare ale indicatorilor economici și stabilirea programului anual definitiv — cît și în faza de ordonanțare — cind se aleg **⟨INTERVAL OPERATIV⟩** și **⟨SUBPERIOADA⟩** de programare și se stabilesc sarcini concrete pe fiecare loc de muncă defalcind o parte din comenzi în cadrul **⟨PROGRAM ANUAL⟩** și adăugind sau scoțind altele, în funcție de condițiile locale, specifice, de la momentul respectiv.

Utilitatea deosebită pe care o are sistemul SOFTPLAN în cadrul conducerii operative a producției în întreprinderile cu profil de construcție de mașini se explică prin avantajele acestui sistem în comparație, de exemplu, cu pachetele de programe de firmă — CLASS (pentru calculatoarele IBM) sau ORDO (pentru calculatoarele FELIX-C512) — și care pot fi rezumate, în principal, la următoarele [37]:

a) Numărul mare de informații necesare funcționării pachetelor de programe de firmă — unele din ele greu de estimat sau de calculat, nefiind utilizate în mod curent în practică — este redus pentru sistemul SOFTPLAN la strictul necesar; Se cer numai duratele operațiilor importante care se execută pe posturile de lucru „cheie” și succesiunea acestora, capacitățile reale ale posturilor, lista comenziilor-produs și a limitelor lor de începere/terminare a fabricației, cu alte cuvinte informații de care se dispune în orice întreprindere și cu ajutorul cărora se face orice fel de programare, chiar într-un sistem în totalitate manual.

b) Rigiditatea în emiterea documentației, lipsa de experiență în adaptarea pachetelor de firmă la nevoile reale ale întreprinderilor noastre, necunoașterea exactă a algoritmului central de ordonanțare, sint înlocuite în cadrul sistemului SOFTPLAN cu o suplețe și adaptabilitate maximă datează modularizării, parametrizării, intervalului de programare ales convențional și mai ales conceperii întregului ansamblu ca un *sistem procedural* ([19] p. 124—134). În cadrul unui astfel de sistem nu există procedură rigidă de funcționare ci numai niște *condiții generale de utilizare* (vezi în cadrul capitolului 4 **⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩**). Tânărind seama de caracterul fazei de programare (**⟨PROGRAM ANUAL⟩** sau un **⟨INTERVAL OPERATIV⟩** din cadrul acestuia), de obiectivele urmărite (evaluare cu caracter general sau programare mai de detaliu), de posibilitățile locale de acoperire ale eventualelor supraîncărcări restante din programare și, în cele din urmă, de experiența personală, fiecare utilizator își construiește propria sa procedură: alege modulul de program potrivit, îi precizează parametrii, îl execută pe calculator și în funcție de rezultatele obținute trece la pasul următor. Fiecare pas poate consta dintr-o actualizare manuală sau o optimizare automată; se obține o ameliorare continuă a rezultatului — fapt demonstrat matematic prin convergența algoritmului de ordonanțare — procesul fiind oprit de către utilizator odată cu atingerea obiectivului urmărit.

c) Programarea de detaliu, rigidă, „la oră” și „pe mașină”, efectuată în general de către pachetele de programe de firmă, nu poate fi pusă în practică — decit cu rare excepții — în întreprinderile noastre construcțoare de mașini, din cauza fluctuației permanente a unor parametri esențiali care intervin în calcule, ca de exemplu: durata operației, timpul de pregătire/incheiere, timpul de transport inter-operații, capacitatea disponibilă etc. De asemenea, în foarte multe întreprinderi intervin o serie de factori imprevizibili, cum ar fi: utilaj defect sau care nu funcționează la parametrii nominali, înlocuirea forței de muncă programate cu o alta, de calificare inferioară, apariția unor comenzi suplimentare urgente, nerespectarea termenului de intrare în fabricație, deficiențe în aprovizionarea cu materiale și semifabricate, rebuturi etc. Ori, pachetele de programe de firmă menționate nu se pot adapta noilor situații în decurs de cîteva ore, astfel încit să nu creeze perturbații importante în producție.

Pachetul de programe SOFTPLAN lucrează cu o perioadă de programare variabilă, de ordinul citorva zile, o săptămînă, decadă, lună etc. și la nivelul unei grupe de mașini care execută aceeași gamă de operații. Ca urmare, numărul informațiilor manipulate se reduce simțitor iar frecvența de actualizare scade, ceea ce ușurează mult întreținerea bazei de date. Pe de altă parte, programarea manoperei pe grupe de mașini la interval de cîteva zile *lăsă liberă inițiativa* conducerii pentru a compensa prin măsuri locale mici perturbații ce apar frecvent în procesul de producție. Rezultatul este anihilarea practic completă a abaterilor mărunte de la program, răminind să fie luate în considerare și reprogramate numai abaterile importante. *Ceea ce urmează să fie pus în practică și controlat este, prin urmare, un program mult mai apropiat de necesitățile reale ale întreprinderii și care se poate realiza cu mijloace relativ modeste.*



Dat fiind interesul pentru introducerea în practica întreprinderilor a unor instrumente de conducere perfecționate de acest tip precum și rolul deosebit al acestor mecanisme în realizarea conceptului de model intelligent despre care am vorbit, vom prezenta pe scurt în cele ce urmează sistemul SOFTPLAN: modelul matematic, realizare informatică precum și unele rezultate semnificative obținute pînă în prezent cu acest sistem în ramura construcțiilor de mașini.

6.2. Modelul matematic

Să considerăm următoarele mulțimi care caracterizează procesul de producție al unei întreprinderi: R = mulțimea comenziilor-produs.

O comandă-produs $r \in R$ se caracterizează prin: QT este cantitatea de produse în cadrul comenzi, PRIO — prioritatea comenzi, TID — timpul de început cel mai devreme posibil al fabricației, TST — timpul de sfîrșit cel mai tîrziu admis pentru fabricație.

U — mulțimea posturilor de lucru.

Un post de lucru $u \in U$ poate fi ales, convențional, drept grupă de mașini, atelier, secție etc.

A — mulțimea articolelor.

Un articol $a \in A$ poate fi un produs, ansamblu, subansamblu, reper sau material care intră în alcătuirea produsului.

STR — mulțimea structurilor articolelor.

Un element $\text{pred}_a \in \text{STR}$ al acestei mulțimi este o pereche ordonată de mulțimi (\bar{a}, Q) referitoare la articolul $a \in A$:

$$\bar{a} = (a_1, \dots, a_n) \quad (6.1)$$

$$Q = (q_1, \dots, q_n) \quad (6.2)$$

în care $a_i \in A$, $i = 1, \dots, n$ preced în execuție pe a , fiind necesare în cantitățile q_i , $i = 1, \dots, n$ la fabricarea lui a iar O – mulțimea operațiilor corespunzătoare fabricării fiecărui articol.

Succesiunea lor în timp este definită de procesul tehnologic.
Pentru un articol $a \in A$ se definește

$$\lambda_a = (o_1, \dots, o_p), \quad o_i \in O \quad i=1, \dots, p. \quad (6.3)$$

lanțul operațiilor tehnologice care conduc la fabricarea articolului.

Fiecare operație o_i îi corespunde o durată a fabricației τ_i , în care am inclus, pentru simplificare, și timpul de transport și cel de pregătire-încheiere, astfel încât putem determina, pentru un λ_a , o durată totală a fabricației articolului a

$$t_a = \sum_{i=1}^p \tau_i. \quad (6.4)$$

Fiecare operație se execută pe un anumit post de lucru și servește la fabricarea unui anumit articol.

Notăm cu $o_{a,u}$ operația corespunzătoare articolului $a \in A$ și postului de lucru $u \in U$.

Se definesc următoarele aplicații:

$$\text{lan}_{RA} : R \rightarrow A \quad (6.5)$$

care asociază unei comenzi-produs $r \in R$, produsul corespunzător $a \in A$ din mulțimea articolelor;

$$\text{lan}_{AS} : A \rightarrow \text{STR} \quad (6.6)$$

care asociază unui articol $a \in A$ elementul $\text{pred}_a \in \text{STR}$;

$$\text{lan}_{AO} : A \rightarrow O \quad (6.7)$$

care asociază unui post de lucru $u \in U$ operațiile care se execută pe acest post;

$$\text{lan}_{oA} : O \rightarrow A \quad (6.8)$$

care asociază unei operații, articolul la care se referă operația respectivă;

$$\text{lan}_{ou} : O \rightarrow U \quad (6.9)$$

care asociază unei operații, postul de lucru pe care se execută operația respectivă.

Pe mulțimea A a articolelor se definește aplicația multivocă (graful arborescent)

$$\Gamma(A_a, \text{arc}_a), \quad a \in A, \quad (6.10)$$

în care arc_a este mulțimea arcelor \overline{aa} , care pleacă din a , a_i fiind definit de $\text{pred}_a \in \text{STR}$.

Din teoria grafurilor [7, 58] rezultă posibilitatea de a calcula pentru fiecare produs $r \in R$ durata minimă de fabricație, ținând seama de graful parțial

$$\Gamma_p(A_r, \text{arc}_a), \quad a \in A_r, \quad (6.11)$$

în care A_r este mulțimea componentelor produsului r , iar

$$p = \text{lan}_{\text{RA}}(r). \quad (6.12)$$

Această durată minimă reprezintă ciclul de fabricație al produsului r .

Fie $O_{r,u}$ ansamblul operațiilor ce se execută pe postul de lucru u relativ la produsul r . Să însumăm duratele operațiilor pe fiecare post de lucru în parte, ținând cont de succesiunea lor în timp. Stabilim de la început o dimensiune a perioadei de timp pentru care facem programarea (trimestru, lună, decadă etc.) precum și numărul n de perioade care constituie orizontul de programare. Pentru o perioadă oarecare k , suma de mai sus reprezintă

$$s_{r,u}^k = \sum_{o \in O_{r,u}^k} t_o, \quad (6.13)$$

unde t_o este durata operației O (în ore \times om) iar $O_{r,u}^k$ — ansamblul operațiilor care se execută în perioada k , pe postul u , relativ la comanda produs r .

Vom denumi această mărime *încărcarea comenzi-produs r pe postul u în perioada k* .

Notăm cu S_r mulțimea încărcărilor referitoare la comanda-produs r .

$$S_r = (s_{r,u}^k), \quad k = 1, \dots, n, \quad u \in U_r, \quad (6.14)$$

în care U_r este mulțimea posturilor care intervin în prelucrarea comenzi-produs r .

Fie TID' — timpul de început cel mai devreme posibil ^{*)} și

TST' — timpul de sfîrșit cel mai tîrziu admis ^{**)'}, ambele referitoare la fabricația comenzi-produs r .

Notăm cu

$$\Delta T_r = \text{TST}' - \text{TID}' \quad (6.15)$$

și cu

$$\delta_r = \Delta T_r - L_r, \quad (6.16)$$

în care L_r este lungimea ciclului de fabricație al produsului r (perioade).

^{*)} Din punctul de vedere al existenței tuturor condițiilor pentru începerea fabricației (aprovisionare cu materii prime și semifabricate, contracte de colaborare, condiții de transport etc.).

^{**)'} Ținând seama de termenele prevăzute în contractele de livrare.

Admitem prin ipoteză că nu putem programa decit comenzi-produs care îndeplinesc condiția

$$\delta_r \geq 0. \quad (6.17)$$

Mulțimea comenzi-produs care îndeplinesc această condiție și care va fi inclusă de decident în lista de comenzi programabile va fi notată de aici înainte cu R .

Asociem lui S_r matricea binară a ocupării, definită astfel

$$M_r = (e_{r,u}^k), \quad k = 1, \dots, n, \quad u \in U_r, \quad (6.18)$$

unde

$$[e_{r,u}^k] = \begin{cases} 1 & \text{dacă } s_{r,u}^k > 0, \\ 0 & \text{dacă } s_{r,u}^k = 0. \end{cases}$$

Tinând seama relația (6.18), L_r se definește cu ajutorul relației

$$L_r = [\max(k), e_{r,u}^k = 1] - [\min(k), e_{r,u}^k = 1]. \quad (6.19)$$

Pentru mulțimea p care satisface relațiile :

$$1 \leq p, \quad [\min(k), e_{r,u}^k = 1], \quad (6.20)$$

$$1 \leq p \leq n, \quad [\max(k), e_{r,u}^k = 1], \quad (6.21)$$

definim următoarea pereche de transformări a matricei S_r în S'_r

Translație -p a lui r sau *devansare* (6.22)

$\text{transl}_{-p,r} : S_r \rightarrow S'_r$ în care $s'_{r,u}^k \in S'_r$

$$s'_{r,u}^k = \begin{cases} s_{r,u}^{k+p} & \text{pentru } k = 1, \dots, n-p, \\ 0 & \text{pentru } k = n-p+1, \dots, n, \end{cases}$$

dacă p satisface relația (6.20), și

Translație p a lui r sau *decalare* (6.23)

$\text{transl}_{p,r} : S_r \rightarrow S'_r$ în care $s'_{r,u}^k \in S'_r$

$$s'_{r,u}^k = \begin{cases} s_{r,u}^{k-p} & \text{pentru } k = p+1, \dots, n, \\ 0 & \text{pentru } k = 1, \dots, p, \end{cases}$$

dacă p satisface relația (6.21)

Pentru postul de lucru $u \in U$ calculăm suma încărcărilor

$$\sigma_u^k = \sum_{r \in R} s_{r,u}^k, \quad (6.24)$$

în care R este mulțimea comenzi-produs.

Considerind cunoscută capacitatea cap_u^k a postului de lucru u în perioada k , putem calcula diferența de încărcare sau *suprasarcina*

$$dif_u^k = \sigma_u^k - cap_u^k. \quad (6.25)$$

Maximul acestei diferențe, pentru postul u , este

$$\max_u = \max(dif_u^k), \quad k = 1, \dots, n, \quad dif_u^k > 0. \quad (6.26)$$

Relativ la toate posturile de lucru, se calculează în final :

$$\text{MAX} = \max(\max_u), \quad u \in U. \quad (6.27)$$

Problema este de a minimiza valoarea **MAX** pentru o structură dată a mulțimii R , ținând cont de TID^r, TST^r și de prioritățile comenziilor-produs r .

În vederea rezolvării problemei sus enunțate sistemul SOFTPLAN realizează cu mijloace informaticе transformările $transl_{\pm p,r}$, în așa fel încit să nu permită mărirea valorii **MAX**, sau a numărului de maxime egale, pe parcursul transformărilor. Așa dar, cu fiecare transformare efectuată se obține o situație de progres față de situația anterioară, din punctul de vedere al obiectivului enunțat.

Se definesc următoarele noțiuni :

1° O încercare de diminuare a lui **MAX** sau de mășorare a numărului de maxime egale se numește *iterație*.

2° O iterare se numește *reușită* dacă realizează una din cele două condiții de mai jos

— $\text{MAX}' < \text{MAX}$,

în care MAX' este valoarea **MAX** după efectuarea transformării, sau, pentru $\text{MAX}' = \text{MAX}$,

— $\mathcal{C}\{(u, k) | \text{dif}_u^k = \text{MAX}\} < \mathcal{C}\{(u, k) | \text{dif}_u^k = \text{MAX}\}$,

în care $\mathcal{C}\{(u, k) | \text{dif}_u^k = \text{MAX}\}$, respectiv $\mathcal{C}\{(u, k) | \text{dif}_u^k = \text{MAX}\}$, reprezintă numărul perioadelor (u, k) pentru care suprasarcina dif_u^k rămîne egală cu **MAX**, după efectuarea transformării, respectiv înainte de transformare.

În caz contrar, iterarea se numește *nereușită*.

3° O iterare reușită duce la transformarea matricii S_r în S'_r .

Mai departe se poate continua încercarea de diminuare la lui **MAX** pornind de la noua situație, sau se poate reveni la situația anterioară.

O iterare nereușită nu provoacă nici un fel de schimbare asupra matricilor S_r , $r \in R$.

4° Se obțin o serie de transformări

$$S \rightarrow S' \rightarrow S'' \rightarrow \dots \quad (6.28)$$

cărora le corespund obligatoriu relațiile

$$\text{MAX} \rightarrow \text{MAX}' \rightarrow \text{MAX}'' \rightarrow \dots \quad (6.29)$$

în care am notat cu $S = (s_{r,u}^k)$, $r \in R$, $u \in U$, $k = 1, \dots, n$, MAX fiind mărimea calculată pentru S cu relația (6.27).

Din punctul de vedere al utilizatorului, este necesar :

- Să se obțină o scădere *cât mai rapidă* a valorii MAX calculate inițial.
- Să se ajungă în final la o valoare MAX care îl *satisfacă* din punctul său de vedere.

Deoarece nu există restricții formulate apriori pentru numărul perioadelor planificate, numărul comenziilor-produs și structura acestora, prioritatea comenziilor, structura capacitaților disponibile etc., vom studia fenomenul reducerii succesive a valorii MAX în cadrul unui model probabilistic, luând ca variabilă aleatoare mărimea ρ_u^{*k} — rezerva de încărcare relativă față de MAX , a unui post de lucru u într-o perioadă k , pe care o definim în felul următor :

Fie ρ_u^k rezerva de încărcare absolută față de MAX a postului de lucru u în perioada k

$$\rho_u^k = \text{MAX} + \text{cap}_u^k - \sigma_u^k. \quad (6.30)$$

Rezerva de încărcare relativă se calculează cu formula

$$\rho_u^{*k} = \frac{\rho_u^k}{\text{MAX} + \text{cap}_u^k} = 1 - \frac{\sigma_u^k}{\text{MAX} + \text{cap}_u^k}. \quad (6.31)$$

Variabila aleatoare ρ_u^{*k} poate lua orice valoare în intervalul $[0, 1]$: $\rho_u^{*k} = 0$ pentru $\text{MAX} \neq 0$, corespunde acelor perioade (u, k) pentru care supraîncărcarea este maximă ; $\rho_u^{*k} = 1$ corespunde perioadelor (u, k) în care se înregistrează goluri de sarcină $\sigma_u^k = 0$.

Pentru $\text{MAX} = 0$, atunci cînd nu mai există suprasarcini, $\rho_u^{*k} > 0$ arată că există rezerve de capacitate față de planul propus.

La un moment oarecare t al procesului descris de relațiile (6.28) și (6.29) să considerăm mulțimea $M = \{(u, k)\}$ a tuturor perioadelor, fiecare perioadă fiind caracterizată printr-o valoare ρ_u^{*k} calculată cu formula (6.31).

Să trasăm curba de repartiție a probabilităților cumulate [40] pentru variabila aleatoare ρ_u^{*k}

$$F(x_i) = P(\rho_u^{*k} < x_i), \text{ pentru diferite valori } x_i, x_i \in [0, 1], \quad (6.32)$$

în care P este probabilitatea ca variabila ρ_u^{*k} să fie mai mică decît x_i ; se calculează împărțind numărul perioadelor (u, k) pentru care $\rho_u^{*k} < x_i$ la totalul perioadelor din M .

Curba de distribuție a probabilităților pentru variabila aleatoare ρ_u^{*k} se trasează unind punctele i calculate cu formula ([40] p. 100) :

$$f(x_i) = \frac{F(x_{i+1}) - F(x_i)}{x_{i+1} - x_i}, \quad i = 0, 1, \dots, n - 1, \quad (6.33)$$

unde n reprezintă numărul de puncte luate în considerare și x_i , x_{i+1} sunt două puncte consecutive pe axa absciselor, $x_{i+1} > x_i$, $F(x_{i+1}) \geq F(x_i)$ deoarece curba de repartiție este, prin definiție, monoton nedescrescătoare.

În figura 5 s-au reprezentat $F(x_i)$ — curbele *a* și $f(x_i)$ — curbele *b* — în patru momente diferite ale procesului: I — corespunde situației inițiale, în care există o singură perioadă cu suprasarcina $\text{MAX}(\rho_u^{*k} = 0)$,

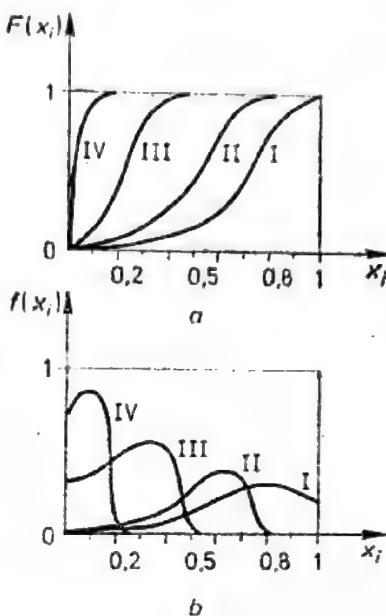


Fig. 5. — Dinamica procesului SOFTPLAN: *a* — curbe de repartiție; *b* — curbe de distribuție; I, II, III, IV — momente succesive ale procesului.

celealte avind rezerve relativ mari, unele chiar maxime ($\sigma_u^k = 0$, $\rho_u^{*k} = 1$); II și III corespund unor stări intermediare în care nu mai există goluri de sarcină ($\sigma_u^k > 0$, $\rho_u^{*k} < 1$); II prezintă tot un singur maxim, III are deja mai multe maxime egale pentru care $\rho_u^{*k} = 0$; IV reprezintă starea finală a procesului, caracterizată printr-o mulțime de maxime (mici) de valoare egală și relativ puține perioade care mai prezintă rezerve de încărcare.

De remarcat faptul că folosind curbele $F(x_i)$ și $f(x_i)$ se pot face și unele aprecieri calitative asupra structurii inițiale a planului, atât global (fig. 6 *a*, *b*) cît și pe fiecare post de lucru în parte (fig. 7 *a*, *b*).

Astfel, în figura 6 *a* și *b*, sănătate reprezentate: *A* — o structură cu un număr aproximativ egal de încărcări de diferite mărimi; *B* — o structură cu un maxim de suprasarcină pronunțat și goluri foarte puține, în rest încărcările fiind relativ echilibrate față de capacitate.

În figura 7 *a* și *b*, sănătate reprezentate: 1 — un post de lucru pe care s-a înregistrat supraîncărcarea maximă (MAX). Postul este neuniform încărcat și conține multe perioade cu goluri de sarcină; 2 — un post de lucru ale cărui suprasarcini sunt mai mici decât cea maximă. Postul este încărcat aproape uniform, dar prezintă rezerve mari față de MAX + cap_u^k ; 3 — idem, ca la punctul 2, rezervele fiind însă mult mai mici (curbele $f(x_i)$ și $F(x_i)$ deplasate mult spre stînga).

Să revenim acum la procesul descris în figura 5.

Din analiza stărilor la momentele I — IV rezultă că, pe măsură ce MAX se micșorează, variabila aleatoare ρ_u^{*k} are, în general, tendință de scădere.

Să considerăm două momente succesive t și t' ale procesului, $t' > t$, $\text{MAX}' \leq \text{MAX}$.

Pentru ca ρ_u^{*k} să seadă, este nevoie ca diferența $\Delta \rho_u^{*k} = \rho_u^{*k} - \rho_u^{*k}$ să fie pozitivă, cu alte cuvinte

$$\Delta \rho_u^{*k} = \frac{\sigma_u^{*k}}{\text{MAX}' + \text{cap}_u^k} - \frac{\sigma_u^{*k}}{\text{MAX} + \text{cap}_u^k} > 0. \quad (6.34)$$

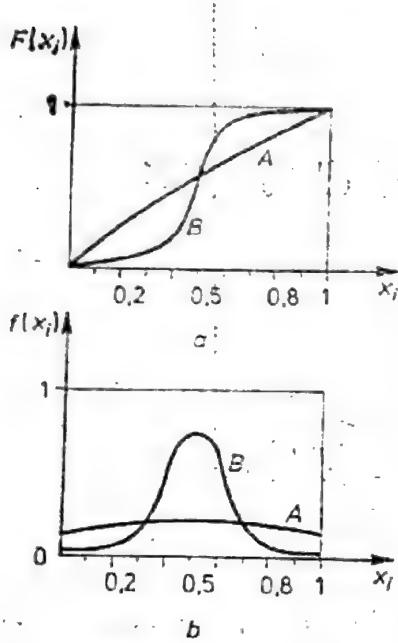


Fig. 6. — Structuri de plan: a — curbe de repartiție; b — curbe de distribuție; A, B — variante de structură.

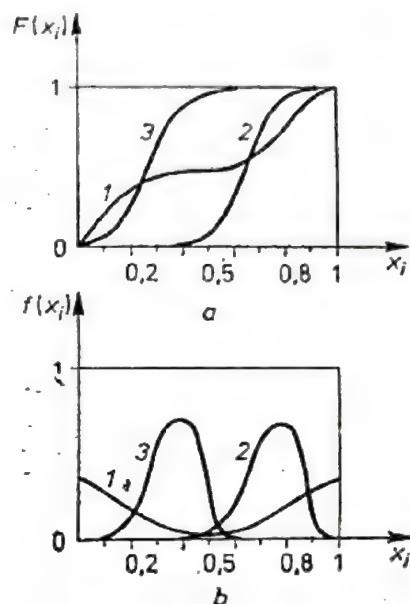


Fig. 7. — Încărcări ale posturilor de lucru: a — curbe de repartiție; b — curbe de distribuție; 1 — post cu maxim; 2 — post fără maxim cu rezervă mare de încărcare; 3 — post fără maxim cu rezervă mică de încărcare.

Notind cu ΔM diferența $\text{MAX} - \text{MAX}'$ și cu $\Delta \sigma_u^k$ diferența $\sigma_u^k - \sigma_u^{*k}$, rezultă relația

$$\frac{\sigma_u^{*k}}{\sigma_u^k} > 1 - \frac{\Delta M}{\text{MAX} + \text{cap}_u^k} \quad (6.35)$$

care, pentru $\Delta M > 0$, mai poate fi scrisă sub forma

$$\frac{\Delta \sigma_u^k}{\Delta M} < \frac{\sigma_u^k}{\text{MAX} + \text{cap}_u^k} = 1 - \rho_u^{*k}. \quad (6.36)$$

Referindu-ne la transformarea $\text{transl}_{\pm p,r}$ (iterație reușită) care deplasează sistemul din starea definită de momentul t în starea definită de momentul t' , diferența $\Delta \sigma_u^k$ mai poate fi scrisă

$$\Delta \sigma_u^k = s_{r,u}^k - s_{r,u}^{k+p}, \quad (6.37)$$

unde p poate fi pozitiv sau negativ.



Pentru ca variabila ρ_u^{*k} să descrească, există, prin urmare, următoarele posibilități :

1°. $\Delta M = 0$ (nu se reduce valoarea MAX, ci se micșorează numărul de maxiime egale).

Din relația (6.35) rezultă $\sigma_u'^k > \sigma_u^k$, ceea ce înseamnă că nu se admit scăderi ale încărcării, ci dimpotrivă, este necesar ca σ_u^k să crească după efectuarea translației.

Acet lucru nu este însă posibil simultan pentru toate încărcările σ_u^k , deoarece suma lor trebuie să rămână constantă.

Rezultă că în acest caz condiția (6.34) nu poate fi, în general, respectată.

2°. $\Delta M > 0$ (valoarea MAX se reduce în urma translației)..

Conform relației (6.36) este necesar ca eventuala scădere a încărcării cu $\Delta \sigma_u^k$ să mențină raportul $\Delta \sigma_u^k / \Delta M$ sub valoarea $1 - \rho_u^{*k}$.

Sunt posibile următoarele cazuri :

a. $\Delta \sigma_u^k = 0$ (este vorba de perioadele care nu sunt afectate de translație).

Relația (6.36) este satisfăcută oricare ar fi valoarea lui ΔM numai pentru $\sigma_u^k > 0$ (perioadele încărcate), deoarece numai pentru acestea $\rho_u^{*k} > 0$.

Perioadele cu $\sigma_u^k = 0$ (goale) nu afectează în vreun fel procesul.

b. $\Delta \sigma_u^k < 0$ adică $\sigma_u'^k > \sigma_u^k$ (încărcarea se mărește după translație).

Relația (6.36) este satisfăcută oricare ar fi valoarea lui ΔM .

c. $\Delta \sigma_u^k > 0$ (încărcarea se micșorează după translație).

Relația (6.36) este cu atit mai ușor de îndeplinit cu cit valoarea lui ΔM este mai mare.

Dar ΔM este cel mai mare la începutul procesului și se micșorează pe măsură ce MAX se micșorează și încărcările se uniformizează, în timp ce $\Delta \sigma_u^k$, calculat cu formula (6.37), rămîne constant de-a lungul procesului ($s_{r,u}^k$ nu variază).

Rezultă, că, pe măsură ce procesul înaintează (valoarea MAX se micșorează), din ce în ce mai multe variabile ρ_u^{*k} nu vor mai îndeplini relația (6.36) și, la un moment dat, procesul se oprește.

În orice caz ΔM nu va putea cobori sub valoarea constantă a celui mai mic $\Delta \sigma_u^k$ calculat cu formula (6.37), deoarece

$$1 - \rho_u^{*k} \leq 1 \Rightarrow \Delta \sigma_u^k \leq \Delta M.$$

Să ne referim acum la întregul proces, descris ca fenomen probabilistic.

Pentru a reprezenta distribuția de probabilitate $f(x_i)$ a variabilei aleatoare ρ_u^{*k} , vom împărți intervalul de variație a lui x într-un număr T de benzi egale, de lățime $1/T$.

Pe mulțimea $M = \{(u, k)\}$ a tuturor perioadelor, definim o partiție $\mathfrak{P}(M)$

$$\mathfrak{P} : M \rightarrow I, \quad I = \{1, 2, \dots, T\} \quad (6.38)$$

unei perioade $(u, k) \in M$ avind

$$(i - 1)/T \leq \rho_u^{*k} < i/T, \quad (6.39)$$

corespunzîndu-i $i \in I$ întreg și pozitiv. I reprezintă deci mulțimea claselor definite pe M de către \mathfrak{D} .

Fie

$$M_i = \{(u, k) \mid \mathfrak{D}(u, k) = i\} \quad (6.40)$$

mulțimea perioadelor care aparțin clasei i și

$$n_i = \text{Card } M_i \quad (6.41)$$

numărul elementelor acestei mulțimi.

Probabilitatea ca variabila aleatoare ρ_u^* să ia o valoare cuprinsă în intervalul i , definit de relația (6.39), este

$$f(x_i) = n_i/m, \quad (6.42)$$

în care

$$m = \sum_{i=1}^T n_i. \quad (6.43)$$

Să calculăm acum valoarea mediei probabilistice [34]. Conform definiției, avem

$$\mu = \sum_{i=1}^T x_i f(x_i), \quad (6.44)$$

în care

$$x_i = i/T, \quad i \in I. \quad (6.45)$$

Înlocuind pe (6.45) și (6.42) în (6.44) și ținînd seama de (6.43), rezultă

$$\begin{aligned} \mu &= \sum_{i=1}^T (i/T) \cdot (n_i/m) = (1/Tm) (n_1 + 2n_2 + \dots + Tn_T) = \\ &= (1/Tm) \left[\sum_{i=1}^T n_i + n_2 + 2n_3 + \dots + (T-1)n_T \right] = \\ &= (1/Tm) \left[2 \sum_{i=1}^T n_i - n_1 + n_3 + 2n_4 + \dots + (T-2)n_T \right] = \quad (6.46) \\ &= (1/Tm) \left[3 \sum_{i=1}^T n_i - n_2 - 2n_1 + n_4 + 2n_5 + \dots + (T-3)n_T \right] = \\ &= (1/Tm) \left[T \sum_{i=1}^T n_i - n_{T-1} - 2n_{T-2} - \dots - (T-2)n_2 - (T-1)n_1 \right] = \\ &= 1 - (1/Tm) \left(\sum_{j=1}^{T-1} n_j + \sum_{j=1}^{T-2} n_j + \dots + \sum_{j=1}^2 n_j + n_1 \right) = 1 - (1/Tm) \sum_{i=1}^{T-1} \sum_{j=1}^i n_j, \end{aligned}$$

care mai poate fi scrisă sub forma

$$\mu = 1 - (1/T) \sum_{i=1}^{T-1} \sum_{j=1}^i (n_j/m) = 1 - (1/T) \sum_{i=1}^{T-1} \sum_{j=1}^i f(x_j). \quad (6.47)$$

Dar

$$\sum_{j=1}^i f(x_j) = F(x_i), \text{ funcția de repartiție,} \quad (6.48)$$

așa încât :

$$\mu = 1 - (1/T) \sum_{i=1}^{T-1} F(x_i). \quad (6.49)$$

Din ultima expresie a lui μ rezultă că *media variabilei aleatoare σ_u^{*k} este măsurată de fapt de către suprafața cuprinsă între curbele $F(x_i) = 1$ și $F(x_{i+1})$.* Din figura 5 a se vede că această suprafață se micșorează pe măsură ce procesul avansează (I-II-III-IV) tinzind către zero — cazul ideal cind toate capacitatele sunt acoperite și nu există suprasarcini.

Pentru două momente succesive t și t' , $t' > t$ ale procesului avem deci

$$\mu' < \mu. \quad (6.50)$$

Scăderea mediei probabilistice reprezintă un indicator sigur de evoluție a procesului în sensul dorit de noi.

În practică însă, este incomod să se calculeze de fiecare dată media pentru toate perioadele $(u, k) \in M$.

Mult mai ușor, și în același timp mai puțin costisitor pentru un algoritm realizat cu calculatorul electronic, ar fi să se calculeze media $\bar{\mu}$ referitoare la mulțimea \bar{M} a perioadelor la care se înregistrează de fapt, în urma translației, o modificare a încărcării σ_u^k , mulțime definită prin

$$\bar{M} = \{(u, k) | \Delta \sigma_u^k \neq 0\}. \quad (6.51)$$

Care este însă legătura între media $\bar{\mu}$ relativ la mulțimea \bar{M} și media μ calculată pentru toate perioadele?

Fie \bar{M} , mulțimea definită de (6.51) și o mulțime oarecare M_I care o include pe \bar{M}

$$M_I = \{(u, k)\}, M_I \supset \bar{M}. \quad (6.52)$$

TEOREMA 1. $M_I \supset M$, și este maximă.

Demonstrație. Într-adevăr, fie un element $(u, k) \in M_I$. El se găsește automat și în M care, prin definiție, cuprinde toate perioadele.

Să presupunem că ar exista o mulțime M^*

$$M^* = \{(u, k)\}, M^* \supset M.$$

Rezultă că ar trebui să existe cel puțin un element $(u, k) \in M^*$ cu proprietatea $(u, k) \notin M$, ori acest lucru este imposibil, deoarece M cuprinde toate perioadele.

TEOREMA 2. Dacă, pentru două momente succesive ale procesului t și t' , $t' > t$, este îndeplinită relația (6.50) pentru mulțimea M , această relație se păstrează oricare ar fi mulțimea M_I , la limită și pentru mulțimea M , cu alte cuvinte :

$$\mu' < \bar{\mu} \Rightarrow \mu'_I < \mu_I, \quad \forall M_I \supset \bar{M}. \quad (6.53)$$

Demonstrație. Să considerăm funcțiile de distribuție și mediile relativ la mulțimile \bar{M} și M_I . Avem, conform relațiilor (6.42), (6.43), (6.46)

$$\bar{f}(x_i) = (\bar{n}_i / \bar{m}), \quad i = 1, \dots, T \quad (6.54)$$

și respectiv

$f_I(x_i) = (n_{Ii} / m_I)$, $i = 1, \dots, T$ funcțiile de distribuție, (6.54') iar :

$$\bar{\mu} = 1 - 1/T \bar{m} \sum_{i=1}^{T-1} \sum_{j=1}^i \bar{n}_j \text{ și} \quad (6.55)$$

respectiv

$$\mu_I = 1 - 1/T m_I \sum_{i=1}^{T-1} \sum_{j=1}^i n_{Ij}, \quad \text{mediile,} \quad (6.55')$$

în care :

$$\bar{m} = \sum_{i=1}^T \bar{n}_i \quad (6.56)$$

și, respectiv

$$m_I = \sum_{i=1}^T n_i, \quad (6.56')$$

reprezintă totalul perioadelor mulțimilor \bar{M} și M_I .

Din ipoteză, pentru mulțimea \bar{M} avem îndeplinită condiția

$$1 - (1/T \bar{m}) \sum_{i=1}^{T-1} \sum_{j=1}^i \bar{n}'_j < 1 - (1/T \bar{m}) \sum_{i=1}^{T-1} \sum_{j=1}^i \bar{n}_j. \quad (6.57)$$

Adăugind în ambii membri ai inegalității numărul perioadelor q , neschimbate după efectuarea translației, obținem, după o serie de operații aritmetice simple

$$1 - (1/T m_I) \sum_{i=1}^{T-1} \sum_{j=1}^i (\bar{n}'_j + q_j) < 1 - (1/T m_I) \sum_{i=1}^{T-1} \sum_{j=1}^i (\bar{n}_j + q_j). \quad (6.58)$$

În baza definiției mulțimilor \bar{M} și M_I rezultă că

$$n_{Ij} = \bar{n}_j + q_j \text{ și } n'_{Ij} = \bar{n}'_j + q_j, \quad j = 1, \dots, T-1, \text{ de unde} \quad (6.59)$$

$$1 - (1/T m_I) \sum_{i=1}^{T-1} \sum_{j=1}^i n'_{Ij} < 1 - (1/T m_I) \sum_{i=1}^{T-1} \sum_{j=1}^i n_{Ij}, \quad (6.60)$$

ceea ce demonstrează teorema 2.

TEOREMA 3. Dacă în urma unei translații se obține pentru \bar{M} o scădere a mediei $\bar{\mu}$ cu $\Delta\bar{\mu}$, produsul $m_I \cdot \Delta\mu_I$ calculat relativ la orice mulțime $M_I \supset \bar{M}$ rămîne constant și egal cu $\bar{m} \cdot \Delta\bar{\mu}$.

Cu alte cuvinte produsul „ $m_I \cdot \Delta\mu_I$ ” este un invariant față de orice mulțime M_I care include mulțimea \bar{M} a perioadelor la care se înregistrează în urma translației o modificare a încărcării σ_u^k .

La limită avem îndeplinită egalitatea

$$\bar{m} \cdot \Delta\bar{\mu} = m \cdot \Delta\mu, \quad (6.61)$$

în care m și $\Delta\mu$ se referă la întreaga mulțime M .

Demonstrație. Fie o mulțime $M_I \supset \bar{M}$. Conform relației (6.46) și utilizând relațiile (6.59) avem

$$\begin{aligned} \Delta\mu_I = \mu_I - \mu'_I &= 1 - (1/Tm_I) \sum_{i=1}^{T-1} \sum_{j=1}^i n_{Ij} - 1 + (1/Tm_I) \sum_{i=1}^{T-i} \sum_{j=1}^i n'_{Ij}, \\ &= (1/Tm_I) \sum_{i=1}^{T-1} \sum_{j=1}^i (\bar{n}'_j - \bar{n}_j). \end{aligned} \quad (6.62)$$

De unde

$$\Delta\mu_I \cdot m_I = (1/T) \sum_{i=1}^{T-1} \sum_{j=1}^i (\bar{n}'_j - \bar{n}_j). \quad (6.63)$$

Membrul drept al egalității (6.63) nu depinde de mulțimea M_I relativ la care s-au calculat $\Delta\mu_I$ și m_I și este egal cu produsul $\Delta\bar{\mu} \cdot \bar{m}$ relativ la \bar{M} , ceea ce demonstrează teorema 3.

Relația (6.61) ne permite să calculăm variația mediei variabilei probabilistice ρ_u^{*k} pentru întreaga structură de plan, atunci cînd cunoaștem variația acestei medii pentru mulțimea perioadelor care se modifică în urma translației

$$\Delta\mu = \bar{m}/m \Delta\bar{\mu}, \quad (6.64)$$

în care $\Delta\mu$ este scăderea mediei pentru întreaga structură de plan, $\Delta\bar{\mu}$ – idem, pentru mulțimea perioadelor care se modifică în urma translației, m – numărul tuturor perioadelor structurii de plan, \bar{m} – numărul perioadelor care se modifică în urma translației.

Studiul efectuat mai sus permite să se tragă unele concluzii referitoare la construcția algoritmului de optimizare. Aceste concluzii sunt următoarele :

1. Este necesară, pe tot parcursul procesului, o corelare a mărimilor $\Delta\sigma_u^k$ și ΔM . Cu alte cuvinte este indicat ca la începutul procesului, cînd ΔM este mare, să se manipuleze acele comenzi care conduc la $\Delta\sigma_u^k$ mari, lăsînd spre sfîrșit cînd ΔM se micșorează, translatarea comenziilor cu $\Delta\sigma_u^k$ mici (vezi 2^o c, p. 123).

2. În vederea scăderii rapide a suprasarcinii maxime MAX, se va proceda în felul următor :

a. Se vor prefera translațiile transl_{pr}, care duc la scăderea maximului MAX și se va alege acea translație care provoacă cea mai mare reducere $\Delta\mu$ a mediei probabilistice a variabilei aleatoare ρ_u^{*k} . Această reducere poate fi calculată cu ajutorul formulei (6.64).

b. Dacă nu există decit translații $\text{transl}_{p,r}$ care duc la scăderea numărului de maxime egale, nu se va utiliza, pentru alegerea comenzi-produs, criteriul mediei probabilistice ca la pct. a, ei criteriul celui mai mare număr de maxime reduse (vezi 1^o, p. 123.).

Cu alte cuvinte se alege acea translație la care diferența ΔN între numărul maximelor înainte de translație și numărul maximelor calculat după efectuarea translației, este maximă.

3. Este inutil să se mai încerce coborîrea valorii MAX atunci cînd s-a ajuns la situația în care ΔM este apropiat de valoarea celui mai mic dintre $\Delta\sigma_u^k$ calculat cu formula (6.37).

Pentru a putea cobori totuși sub valoarea menționată mai sus, se va proceda, dacă este posibil, la „spargerea” unor comenzi produs în mai multe comenzi așa-zise „interne”, diminuind $s_{r,u}^k$. Dacă în urma acestei operații se obține (vezi formula (6.37)) o scădere corespunzătoare a limitei $\Delta\sigma_u^k$, procesul de scădere a maximului supraîncărărilor MAX prin deplasarea comenzi-produs r în limitele TID^r – TST^r poate continua cu succes.

6.3. Arhitectura sistemului

Pe baza concluziilor de mai sus au fost realizate pe calculatorul electronic trei categorii de module de optimizare a încărcării capacitaților de producție :

a. Module care translatează (devansează sau decalează) comenzi-produs r , realizind o scădere rapidă a valorii MAX. În acest scop se utilizează criteriul mediei probabilistice a variabilei aleatoare ρ_u^{*k} pentru $\Delta M > 0$ și criteriul celui mai mare număr de maxime reduse ΔN pentru $\Delta M = 0$. Logica generală a acestei categorii de module este următoarea :

- ||Citește lim|| \square
- 1. ||Calculează MAX||
- Dacă MAX < lim atunci*
- 'LIMITA MAXIMA ATINSA'
- Mergi la 2*
- \square
- i, j \leftarrow 0*
- Pentru toate $r \in R$ | transl _{$\pm p,r$} posibilă*
- ||Efectuează transl _{$\pm p,r$} ||
- Dacă ||iterație reușită|| atunci*
- Dacă $\Delta M > 0$ atunci*
- ||Calculează $(\Delta\bar{\mu}, \bar{m})$ ||
- $i \leftarrow i + 1$*
- $\Delta\mu_i \leftarrow \Delta\bar{\mu}$*
- $p_i \leftarrow p$*
- $r_i \leftarrow r$*



□ *Dacă $\Delta M = 0$ atunci*

 ||Calculează (ΔN) ||

$j \leftarrow j + 1$

$\Delta N_j \leftarrow \Delta N$

$p_j \leftarrow p$

$r_j \leftarrow r$

 □

 □ *||Efectuează $\text{transl}_{\pm p, r}$ ||*

□ *Dacă $i > 0$ atunci*

 ||Calculează $(\max(\Delta \mu_k), k = 1, \dots, i, k^*)$ ||*)

 ||Efectuează $\text{transl}_{p_{k^*}, r_{k^*}}$ ||

Mergi la 1

□

Dacă $i = 0, j > 0$ atunci

 ||Calculează $(\max(\Delta N_k), k = 1, \dots, j, k^*)$ ||

 ||Efectuează $\text{transl}_{p_{k^*}, r_{k^*}}$ ||

Mergi la 1

□

'OPTIMIZAREA TERMINATA'

2. ||Listează $(S_u), u \in U_R$ ||**)

STOP

b. Module care translatează (devansează sau decalează) comenzi-produs r , încercând o divizare a comenziilor r în limitele indicate de către programator.

Logica generală a acestor module este următoarea :

||Citește : lim, div||□

1. ||Calculează MAX||

Dacă MAX < lim atunci

'LIMITA MAXIMA ATINSA'

mergi la 2

□

pentru toate $r \in R | \text{transl}_{\pm p, r}$ posibilă

 ||Efectuează $\text{transl}_{\pm p, r}$ ||

dacă ||iterație reușită|| atunci

mergi la 1

 □

 ||Efectuează $\text{transl}_{\mp p, r}$ ||

□

Pentru toate $r \in R | \text{diviziune posibilă}$

 ||Efectuează diviziune $r \rightarrow r', r''$ ||

 ||Efectuează $\text{transl}_{\pm p, r'}$ ||

) k^ este indicele pentru care are loc maximul.

**) Cu U_R s-a notat mulțimea posturilor de lucru afectate de translațiile $\text{transl}_{\pm p, r}$.

Dacă $\| \text{iterație reușită} \|$ atunci
mergi la 1

$\| \text{Efectuează transl}_{\text{+p},r} \|$
 $\| \text{Efectuează revenire } r', r'' \rightarrow r \|$

'DIVIZAREA TERMINATA'

2. Listeză (S_u) , $u \in U_r$

STOP

O comandă r este divizată în două comenzi r' și r'' .

Condiția de divizare posibilă a unei comenzi este fixată de programator prin valoarea priorității comenzi:

- dacă prioritatea este 0 — comanda nu poate fi translatată și nici divizată;
- dacă prioritatea este 1 — comanda poate fi translatată dar nu se admite divizarea ei;
- dacă prioritatea „prio” este mai mare ca 1, comanda poate fi translatată și, eventual, divizată de un număr de ori egal cu diferența „div-prio”.

Cu fiecare divizare reușită, prioritatea crește cu o unitate.

Divizarea nu se mai poate efectua dacă prioritatea a ajuns la valoarea parametrului „div” stabilit de programator.

Fărimițarea extremă a comenziilor este evitată prin calculul unui optim de divizare. Se elimină divizările care nu duc la o reducere substanțială a supraîncărcării maxime.

Comenziile sunt divizate în funcție de numărul de produse identice care le compun, cele două comenzi r' și r'' care rezultă putând avea cantități diferite dar suma lor să reconstituie comanda inițială.

Comenziile alcătuite dintr-un singur produs nu se pot diviza.

c. Module care încearcă numai devansarea ($\text{transl}_{-p,r}$) sau numai decalarea ($\text{transl}_{+p,r}$) comenziilor-produs.

Logica acestei categorii de module este următoarea :

'Citește lim'

1. 'Calculează MAX'
dacă MAX \leq lim atunci
 'LIMITA MAXIMA ATINSĂ'
mergi la 2

'Pentru toate $r \in R$ | $\text{transl}_{-p,r}$ posibilă'
 $\| \text{Efectuează transl}_{-p,r} \|$
dacă $\| \text{iterație reușită} \|$ atunci
mergi la 1

$\| \text{Efectuează transl}_{+p,r} \|$

'DEVANSAREA TERMINATA'

2. 'Listeză (S_u) , $u \in U_r$ '

STOP



Modulele din această categorie se folosesc, de regulă, atunci cind se încearcă devansarea sau decalarea unor anumite comenzi (puține la număr) într-o ordine strictă de prioritate.

Pe lîngă programele de optimizare despre care am amintit sistemul mai conține : module de creare, completare și actualizare a bazei de date, cu intrarea pe cartele perforate sau bandă magnetică ; module de listare a încărcărilor de manoperă pe comenzi și pe posturi, însotite de bilanțuri capacitate-încărcare înainte și după optimizare ; module care permit trasarea automată a curbelor de încărcare pe dispozitive trasatoare de curbe *graph-plotter* de tip BENSON.

Baza de date de tip SCF, cu fișiere înlănuite, se construiește cu ajutorul unui generator de funcții de concepție originală [38]. Procedurile folosite la generarea sistemului sunt complet automate. Prin precizarea unor parametri se creează posibilitatea de a lucra direct cu codurile existente în întreprinderi, fără a mai fi nevoie să schimbați lor sau introducerea unor modificări. De asemenea, menționăm posibilitatea de a transmite și recepta la distanță datele continute în baza de date precum și de a comanda execuția programelor prin intermediul unor terminale de tip mașină de seris sau *display* legate la o rețea de teleprelucrare.

6.4. Unele rezultate și concluzii obținute prin aplicarea sistemului SOFTPLAN în întreprinderi

Sistemul descris mai sus a fost experimentat pînă în prezent într-o serie de întreprinderi constructoare de mașini și metalurgice precum și la programarea activității de reparație a utilajelor grele din construcții.

Astfel, la IMUA-București au fost programate 48 de comenzi-produs pe 598 posturi de lucru (grupe de mașini-unelte universale), reprezentînd circa 80% din întreaga producție a întreprinderii.

Au fost parcursă trei etape de lucru :

În prima etapă, pregătirea, s-a generat sistemul și s-a încărcat baza de date, efectuîndu-se și unele ajustări (manuale) ale cielurilor de fabricație.

În cea de-a două etapă s-au utilizat succesiv două module de optimizare automată a încărcării posturilor de lucru : primul (SIM1B) cu scădere rapidă a suprasarcinii fără divizare, cel de-al doilea (SIM2A) cu divizarea comenziilor și redistribuirea lor în intervalul de programare TID-TST. Rezultatul a fost anularea aproape totală a supraîncărcărilor inițiale. Pentru marea majoritate a posturilor s-a obținut la sfîrșitul acestei etape o încărcare echilibrată, uniformă, fără suprasarcini (vezi de exemplu în fig. 8a, postul 14029944 reprezentînd o mașină prelucrătoare specială utilizată la maximum, în 3 schimburi, și în fig. 8b postul 40090110 din categoria montaj-ajustaj, avind capacitatea variabilă în funcție de numărul echipelor ocupate, a cărui încărcare finală permite o foarte bună utilizare a forței de muncă).

Cea de-a treia etapă a constat în unele mici echilibrări manuale care s-au efectuat în final la cele cîteva posturi de lucru care mai prezențau supraîncărcări (vezi de exemplu în fig. 9, suprasarcina de cîteva sute de ore rămasă neacoperită în lunile 6-7 și 9-10 la postul 31021441 a fost preluată de către postul echivalent 31021921 din cadrul aceleiași secțiîi).

Cele 18 comenzi inițiale s-au transformat, prin divizare automată, în 48 comenzi-produs fiind în așa fel distribuite pe parcursul anului încit să acopere uniform capacitatele disponibile (vezi de exemplu în fig. 10,

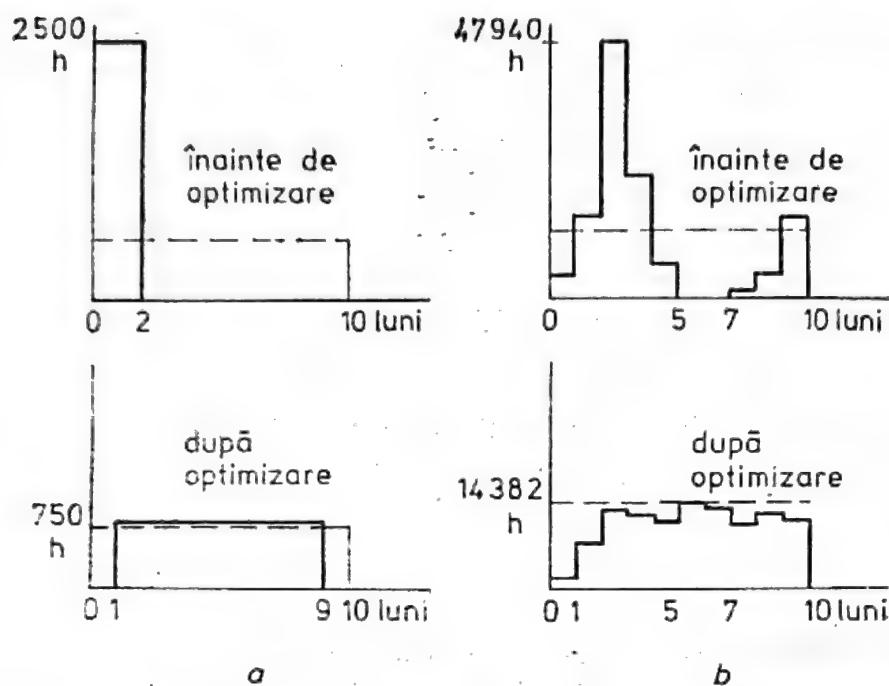


Fig. 8. — Posturi înainte și după optimizarea automată (IMUAB):
a — postul 14029944 (uzinaj); b — postul 40090110 (montaj);
— — fond de ore disponibil; — încărcare lunară (ore/lună).

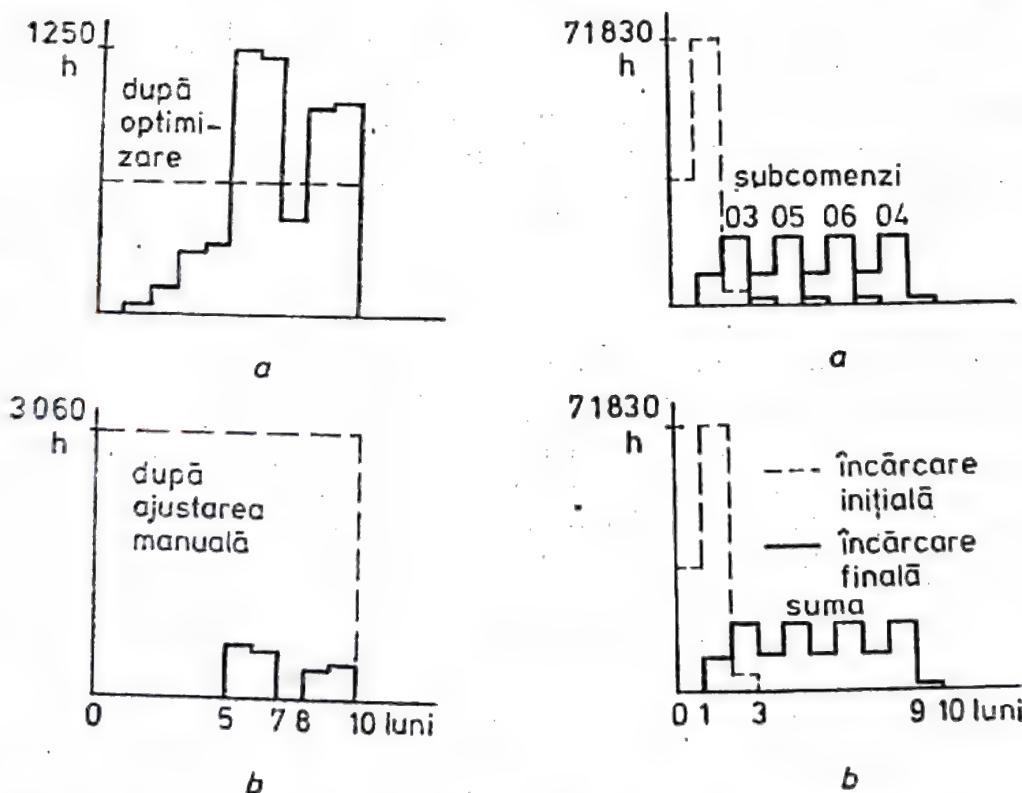


Fig. 9. — Ajustarea finală (IMUAB): a — postul 31021441; b — postul 31021921.

Fig. 10. — Divizarea comenziilor (IMUAB): a — subcomenzi create de calculator; b — suma subcomenziilor.

comandain inițială 23 600 divizată în patru subcomenzi : 03, 05, 06, 04, distribuite în intervalul 1÷10).

Procesul a durat circa 13 ore pe calculatorul electronic **FELIX-C256** fiind executat în 8 pași.

Programul de lansare obținut pentru aceste comenzi a corespuns pe deplin necesităților întreprinderii.

★

La Întreprinderea „Electrotehnica” — București s-au programat, un număr de 30 comenzi pe 9 posturi de lucru principale de uzinaj și montaj. Comenziile au fost selecționate pentru acest experiment cu ajutorul unei diagrame de tip „ABC” (Pareto) și reprezintă circa 75% din producția valorică a întreprinderii.

Pentru calculul ciclogramelor s-a aplicat următoarea metodă :

1°. Pe baza fișelor tehnologice s-a calculat, pentru fiecare produs, încărcarea pe post de lucru în ore manoperă și s-a înmulțit cu numărul de produse al comenzi, rezultând repartizarea totală a manoperei pe comandă și post de lucru.

2°. S-a comparat această încărcare cu capacitatea medie a posturilor. Acolo unde încărcarea unei comenzi era prea mare și ocupa peste 50% din capacitatea medie lunară a postului, s-a divizat (manual) comanda inițială într-un număr de subcomenzi interne capabile de a fi executate eșalonat, fără a amenința blocarea posturilor. De exemplu, postul de lucru PR112 (presaj) are o capacitate medie de 4 000 ore/lună iar comanda IRG2 (100 buc. pe an) încarcă acest post cu 2 500 ore/an. Dacă ar fi executată într-o singură lună, această comandă ar bloca postul PR112 în proporție de 62,5%. Din experiență s-a stabilit că o încărcare de 625 ore/lună ar fi acceptabilă pentru acest tip de produs și în consecință comanda inițială a fost transformată în 4 subcomenzi interne care au fost răspândite pe parcursul anului, așezarea lor finală urmând să se facă în urma optimizării cu ajutorul pachetului de programe **SOFTPLAN**.

3°. Nu au fost divizate comenziile mici, care nu puneau în pericol blocarea posturilor.

Procedind ca mai sus, s-a efectuat o încărcare inițială a posturilor cu toate comenziile-produs. Bilanțul a arătat o serie de supraincărcări, variind între 4 500 ore/lună și 11 500 ore/lună, cele mai mari fiind înregistrate la postul CM131 (construcții metalice — vezi fig. 11). În această fază s-a trecut la aplicarea programelor de optimizare **SOFTPLAN** care au efectuat redistribuirea comenziilor interne, în aşa fel încit în final supraincărcarea maximă a fost micșorată la numai 3 750 ore/lună la postul CM131 (o reducere de 7 750 ore/lună), situația ameliorându-se substanțial și la celelalte posturi (vezi de exemplu în fig. 11 postul TR124). Optimizarea a durat aproximativ 1 oră pe calculatorul electronic.

În urma aplicării metodei de mai sus a rezultat programul de lansare în fabricație pentru 59 de comenzi interne construite și eșalonate cu ajutorul calculatorului, comenzi care conțin principalele produse ale întreprinderii (vezi tabela 4).

★

În vederea programării și controlării producției, la Întreprinderea de Aparate Electronice de Măsură și Industriale (IEMI) s-a aplicat sistemul **SOFTPLAN**. Dat fiind numărul relativ mare de produse de același

Tabela 4 continua

	1	2	3	4	5	6	7	8	9	10	11	12	13
COM 200													
COM 201													
COM 210													
COM 220													
COM 230													
COM 240													
CCM 250													
COM 260													
COM 261													
COM 270													
COM 271													
COM 280													
COM 290													
COM 300													
COM 301													
COM 302													
COM 303													

fel precum și uniformitatea cererii, s-a avut în vedere asigurarea unei ritmicități constante a producției pe tot parcursul anului. Ca atare, producția anuală a fost împărțită în cantități egale pe fiecare lună și fiecare

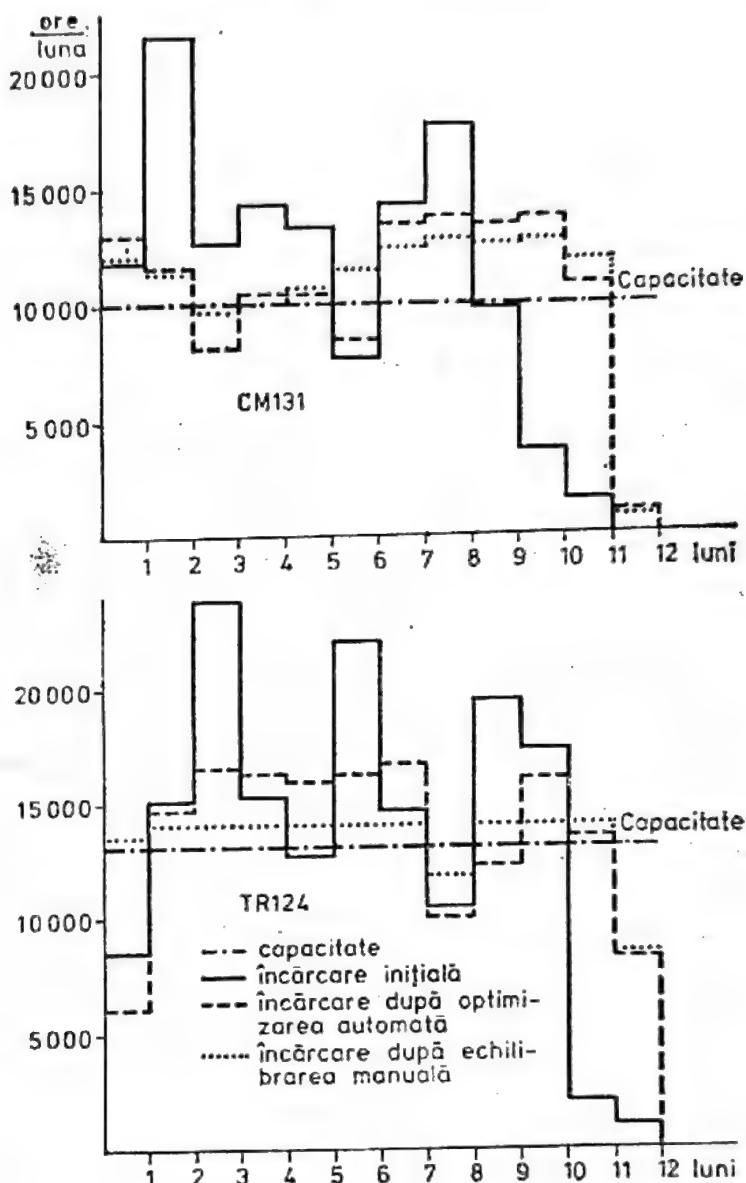


Fig. 11. — Optimizarea încărcării posturilor CM131 și TR124 (Întreprinderea Electrotehnica).

tip de produs, programul de producție lunar urmând a fi optimizat cu ajutorul calculatorului.

Perioada de lucru programabilă a fost stabilită săptămîna. Pentru a oferi o elasticitate mai mare sistemului de urmărire, producția a fost programată pe 6 săptămîni consecutive, primele două aparținind lunii trecute iar următoarele patru — lunii în curs, reprogramarea rămînînd să se facă săptămînal sau lunar, prin luarea în considerare atît a restanțelor cît și a posibilităților de devansare a unor comenzi din perioadele următoare.

În cadrul experimentului au fost încărcate în baza de date un număr de 17 comenzi-produs. Posturile de lucru (grupe de mașini) au fost numerotate simbolic de la 10001 la 10012, pentru acestea calculindu-se o

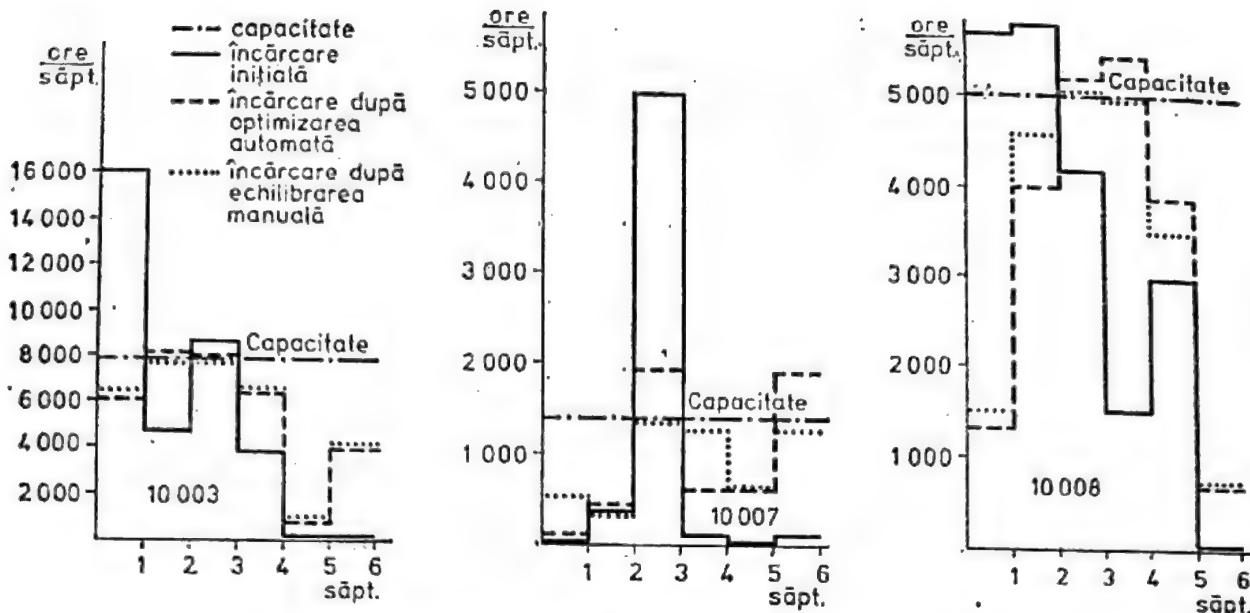


Fig. 12. — Optimizarea încărcării posturilor 10003, 10007 și 10008 (IEMI).

capacitate medie, constantă în fiecare lună. Avindu-se în vedere ciclul real de fabricație (între 1 și 3 săptămâni) comenziile au fost încărcate de posturile de lucru în intervalul $1 \div 6$ săptămâni.

Procesul de optimizare s-a desfășurat în trei faze :

a. S-a aplicat modulul SIM1B-SOFTPLAN încărcării inițiale, ceea ce a dus la o scădere a suprasarcinii maxime de la 8 467 ore (în perioada 1 la postul 10003 — strunguri paralele), la 1 817 ore (în perioada 3 la aceeași post).

b. Pentru a putea micșora și mai mult nivelul suprasarcinilor a fost nevoie de divizarea comenziilor inițiale. În acest scop, s-a aplicat modulul SIM2A-SOFTPLAN, ajungîndu-se în final la o supraîncărcare maximă de numai 536 ore (în perioada 6 pe postul 10007 — mase plastice); s-a limitat numărul divizărilor posibile pentru aceeași comandă la 3, rezultînd în final un număr de 30 comenzi-produs interne, construite de calculator.

c. Supraîncărcările restante au fost anulate printr-o serie de deplasări manuale, în cadrul limitelor permise de tehnologie și ținînd cont de condițiile concrete de lucru.

Pentru ilustrarea rezultatelor, dăm alăturat (vezi fig. 12) imaginea încărcărilor a) inițială, b) după utilizarea pachetului de programe SOFTPLAN și c) după echilibrarea finală manuală pentru 3 din cele 11 posturi (grupe de mașini) pentru care s-a efectuat programarea : postul 10003, postul 10007 și postul 10008. Se observă, în toate cele trei cazuri, aportul substanțial al modulelor de optimizare SOFTPLAN la uniformizarea încărcării posturilor.

*

La întreprinderea Electromureș din Tg. Mureș pachetul de programe SOFTPLAN a fost utilizat în vederea optimizării programului lunar de fabricație al secției cablaje electrice în aşa fel încît să se asigure o încărcare cît mai uniformă a celor patru celule de lucru fundamentale :

— vopsire-debitare-dezisolare conductori (postul DEBIT),

- sertizare elemente terminale metalice (postul SERTZ),
- montat cablaje pe șabloane (postul MSABL),
- rest montaj (postul RESTM).

Succesiunea operațiilor pe cele patru posturi este obligatorie în ordinea menționată mai sus.

Procesul de optimizare cu ajutorul pachetului de programe SOFTPLAN a cuprins următoarele etape :

1. Calculul manoperei necesare pentru fiecare post.

Această operație s-a executat pe baza unor fișe de montaj special proiectate.

Experimentul a fost efectuat pentru 137 tipuri de cablaje, cu cea mai mare pondere în secție, reprezentând circa 20% din planul secției pe luna iulie 1980.

Listele totalizatoare au fost elaborate cu ajutorul calculatorului, la oficiul de calcul al întreprinderii.

2. Perforarea cartelelor și încărcarea fișierelor SOFTPLAN.

Fiecare comandă-produs (tip de cablaj executat într-un număr de bucăți) a fost repartizată inițial în mod aleator în una din cele 4 săptămâni pentru care s-a efectuat programarea. În acest scop, s-au perforat cartele de creare a fișierelor și s-au utilizat programele de încărcare.

3. Optimizarea programului de lansare în fabricație.

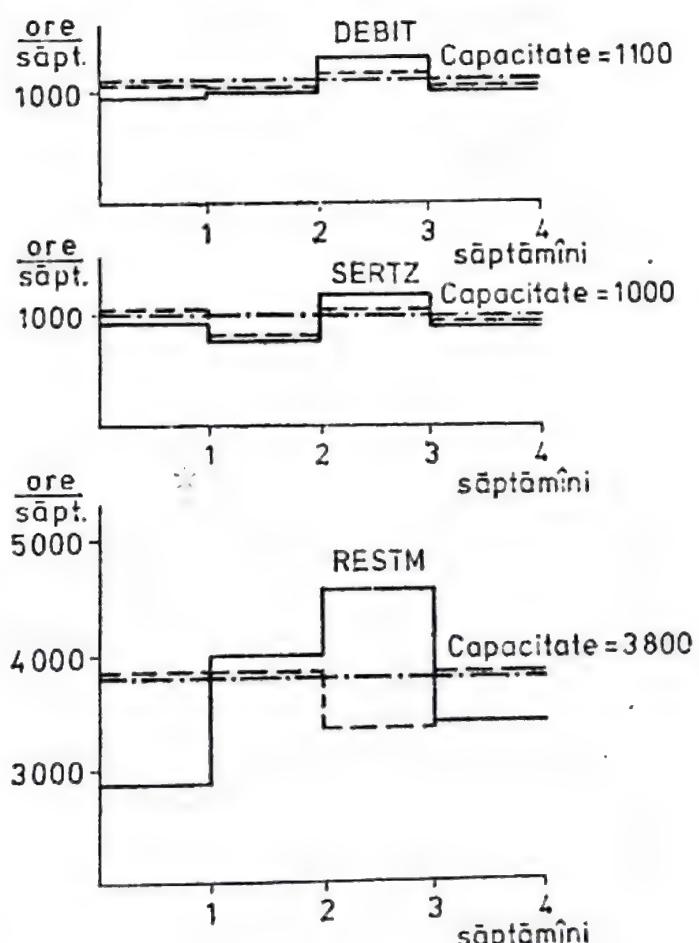


Fig. 13. — Încărcarea posturilor DEBIT, SERTZ, și RESTM, înainte și după optimizare (Întreprinderea Electromureș).

Utilizând modulul SIM1B-SOFTPLAN s-a efectuat în cele din urmă optimizarea încărcării posturilor de lucru DEBIT, SERTZ, în așa fel încât comenziile să fie repartizate cât mai uniform (vezi tabela 5 și fig.13). După

circa 1/2 ore de rulare pe calculatorul electronic s-au stabilit datele de lansare în fabricație pentru cele 137 de comenzi. De remarcat faptul că cele mai mici comenzi au fost neglijate în această programare, ele urmând a fi lansate pe parcurs, din dispoziția șefului de secție, pentru a umple mai bine golurile restante precum și cele create în urma unor indisponibilități temporare.

Tabela 5

**BILANȚUL CAPACITATE-ÎNCĂRCARE ÎNAINTE ȘI DUPĂ OPTIMIZARE
CU AJUTORUL „SOFTPLAN” LA ÎNTREPRINDEREA „ELECTROMUREȘ”**

Postul de lucru		Săptămîna			
		1	2	3	4
DEBIT	Capacitatea postului (ore/săpt)	1 100	1 100	1 100	1 100
	Încărcarea înainte de optimizare (ore/săpt)	977	1 013	1 290	1 013
	Încărcarea după optimizare (ore/săpt)	1 097	1 034	1 127	1 044
SERTZ	Capacitatea postului (ore/săpt)	1 000	1 000	1 000	1 000
	Încărcarea înainte de optimizare (ore/săpt)	908	828	1 158	935
	Încărcarea după optimizare (ore/săpt)	1 019	844	1 003	963
ESTM	Capacitatea postului (ore/săpt)	3 800	3 800	3 800	3 800
	Încărcarea înainte de optimizare (ore/săpt)	2 851	3 999	4 573	3 390
	Încărcarea după optimizare (ore/săpt)	3 824	3 825	3 336	3 828



Sistemul SOFTPLAN a fost experimentat și pentru planul de turbine al I.M.G.B. În cadrul experimentului au fost parcurse următoarele etape :

1. *Alcătuirea fișierului de comenzi.* Având în vedere ciclul lung de fabricație al turbinelor și faptul că fiecare turbină consumă un volum important de ore manoperă, s-a programat fiecare produs în cadrul unei comenzi independente. Din cele 6 tipuri de turbine au fost alcătuite în acest fel 13 comenzi-produs.

2. *Alcătuirea fișierului posturi de lucru.* Au fost luate în considerare 11 posturi de lucru (grupe de mașini) a căror capacitate disponibilă în fiecare lună a fost calculată în funcție de numărul de mașini și ținând cont de perioadele de reparații planificate.

3. *Alcătuirea ciclogramelor de fabricație.* Pentru calculul ciclogramelor s-a procedat mai întii la descompunerea fiecărui produs în ansambluri mari componente : rotori și carcase de înaltă, medie și joasă presiune. Pentru fiecare ansamblu s-a alcătuit apoi cîte o diagramă de flux (vezi

fig. 14) în care s-a reprezentat, conform tehnologiei de fabricație, succesiunea posturilor și numărul de ore manoperă necesar executarea fiecărui grup de operații pe post. Fluxurile au fost apoi decalate corespun-

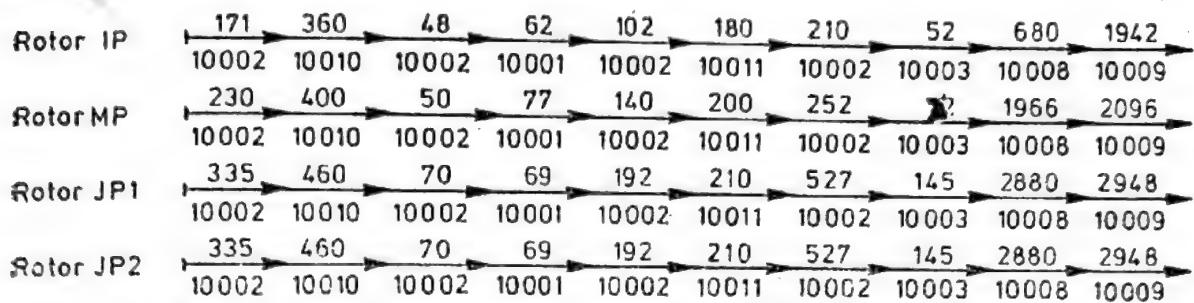


Fig. 14. — Diagramme de flux pentru rotorii turbinei de 330 MW (IMGB).

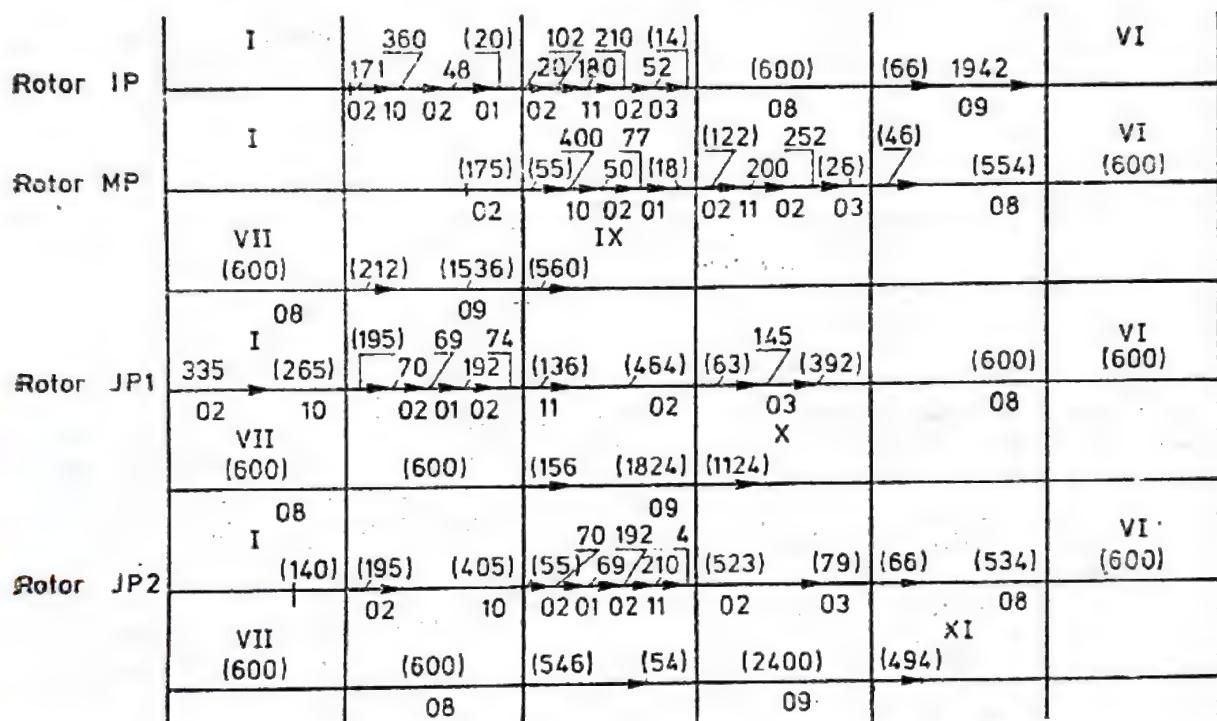


Fig. 15. — Calculul ciclogramelor pentru turbina de 330 MW (IMGB).

zător ordinii de montaj și încadrate în perioade programabile (vezi fig. 15). Prin insumarea orelor de manoperă pe fiecare post de lucru și perioadă au rezultat în cele din urmă ciclogramele de fabricație. Informațiile finale au fost perforate pe cartele servind la crearea fișierului de încărcări.

4. *Optimizarea pe 12 perioade.* Încărcarea inițială a comenziilor în intervalul ianuarie-decembrie al anului 1981 în conformitate cu planul dat a pus în evidență existența unor supraîncărcări foarte mari pe posturile de lucru, ceea ce face imposibilă realizarea planului la termenele stabilite și cu tehnologia de producție actuală. Încercarea de optimizare pe 12 perioade s-a lovit de imposibilitatea diminuării acestor suprasarcini, constante pe întreg parcursul anului.

5. *Optimizarea pe 18 perioade.* Pentru a oferi totuși o soluție realizable s-a trecut la programarea acelorași comenzi în intervalul 1-18 (pe un an și jumătate).

În urma aplicării modulului SIM1B-SOFTPLAN supraîncărcările inițiale maxime pe fiecare post de lucru au fost diminuate (vezi fig. 16) rezultând o repartizare mult mai bună a manoperei de-a lungul intervalului ales.

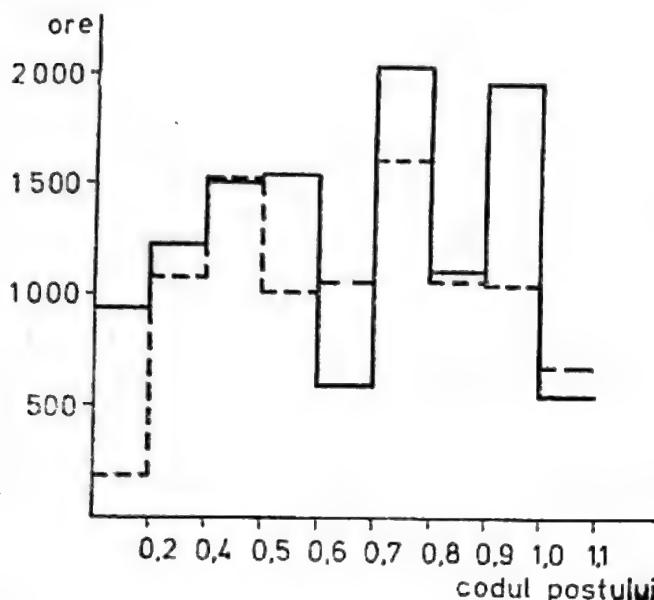


Fig. 16. — Situația încărcărilor maxime pe posturi, înainte și după aplicarea SOFTPLAN (IMGB):
— înainte de aplicare ;
- - - după aplicare.

6. *Ajustarea manuală finală.* Având în vedere unele modificări posibile de tehnologie precum și disponibilul de capacitate în unele perioade, s-a reușit ca în final, prin unele devansări sau decalări manuale pe situațiile oferite de calculator, să se asigure realizarea programului de fabricație. Astfel, pentru postul 10002, cele 187 ore supraîncărcare din luna 2-a și 105 ore din luna 4-a pot fi realizate în luna 3-a ; la postul 10004, 1137 ore suprasarcină din luna 6-a pot fi anihilate prin prelucrarea comenzi C3 în lunile 2, 3 și 4 ; postul 10005 are o suprasarcină de 1531 ore în luna 10-a și diferite supraîncărcări în celelalte luni, care se pot rezolva prin schimbarea tehnologiilor de prelucrare pentru comenziile C2 și C5 pe caruselul de 16 000 mm ș.a.m.d.

Avantajele utilizării sistemului SOFTPLAN sunt evidente și rezultă din folosirea mai bună a capacitaților de producție existente precum și din ușurința cu care pot fi luate din timp deciziile pentru rezolvarea unor cazuri mai dificile. În final se stabilește programul de execuție care asigură livrarea la timp a produselor.



În vederea rationalizării operațiilor de construcție a navelor și scurătării ciclurilor de fabricație, la Întreprinderea de Construcții Navale și Prelucrării la Cald (ICNPC) — Drobeta Turnu-Severin s-au întocmit grafuri-rețea care au constituit baza de calcul pentru ciclogramele necesare aplicării pachetului de programe SOFTPLAN. S-a folosit următoarea metodă : În partea inferioară a grafului a fost adăugat un caroaj care păstrează pe orizontală desfășurarea în timp a operațiilor (putind fi împărțit în perioade de programare, de exemplu luni), iar pe verticală conține lista posturilor de lucru (meserilor) pe care se execută operațiile

(vezi fig. 17). Pentru fiecare operație trecută în graf rezultă un număr de ore necesar pe diverse meserii. Totalizând aceste ore, pe meserii, pentru fiecare perioadă programabilă (lună) rezultă ciclogramele de încărcare pe post de lucru și perioadă, care se utilizează de către SOFTPLAN.

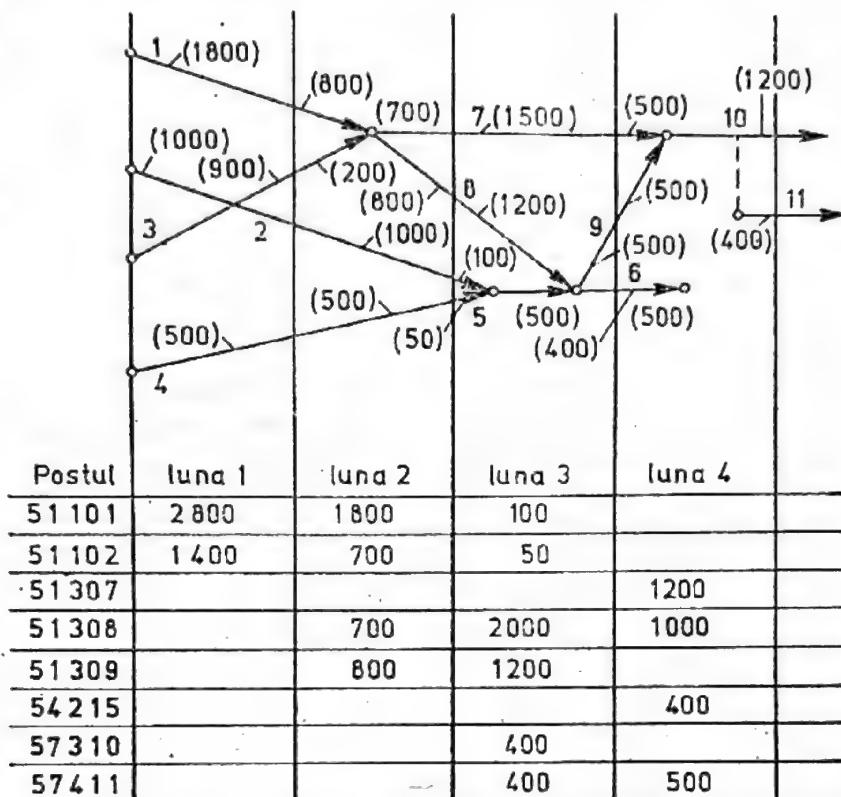


Fig. 17. — Graful operațiilor și calculul ciclogramelor de fabricație ale unci nave (ICNPC Drobeta Turnu Severin): 1 — debitare (51101) profile corp navă ; 2 — debitare (51101) profile puncte superioară ; 3 — croire tablă (51102) corp navă ; 4 — croire tablă (51102) puncte superioară ; 5 — turnare suporti (57310) puncte superioară ; 6 — forjare (57411) arbore ; 7 — sudură electrică (51308) ansamblul I corp navă ; 8 — sudură gaze (51309) ansamblul I corp navă ; 9 — sudură electrică (51308) ansamblul II corp navă ; 10 — montaj (51307) compartimentele I—IX ; 11 — ajustaj (54215) compartimentele I—IX...

Experimentarea sistemului SOFTPLAN s-a efectuat pornind de la planul director primit de la Centrală. S-a acționat în două etape: (a) Încărcarea inițială, (b) optimizarea, în mai multe variante, cu ajutorul calculatorului electronic.

(a) Încărcarea inițială a urmărit strict indicațiile planului director. Rezultatul a scos în evidență mari strangulări de capacitate, acestea ajungind în unele luni la peste 78 000 ore manoperă. Se remarcă, de asemenea, lipsa de capacitate constatătă în mod constant, de-a lungul anului, la unele meserii ca: electricieni, tubulatori, ajustori, lăcațuși etc.

(b) După aplicarea pachetului de programe SOFTPLAN, menținind restricțiile de termene din planul director, s-a obținut o primă variantă de programare în care fluctuațiile de încărcare s-au mai ameliorat, vîrful de suprasarcină scăzînd de la 78 000 ore/lună la circa 60 000 ore/lună, ceea ce reprezintă un cîștig în forță de muncă utilizată de 18 000 ore (circa 90 muncitori) la ajustaj, 3 000 ore (circa 15 muncitori) la electri-



cieni, 5 000 ore (circa 25 muncitori) la lăcătuși etc. Dată fiind menținerea supraîncărcărilor pe tot parcursul anului, s-a încercat obținerea unei a 2-a variante, prin prelungirea unor comenzi pînă la jumătatea anului următor. Aceasta a dus la scăderea suprasarcinii maxime pînă la aproape 40 000 ore/lună ceea ce reprezintă o reducere a forței de muncă utilizate în unele luni cu circa 38 000 ore (190 muncitori). De menționat că reducerea s-a obținut de astă dată cu prețul prelungirii termenelor de dare în funcțiune la unele nave, lucru care, de altfel, avind în vedere experiența anilor trecuți, era de așteptat.

În urma experimentării pachetului de programe SOFTPLAN la ICNPC Drobeta Turnu-Severin au fost scoase în evidență următoarele constatări importante :

1. Supraîncărcările mari constatate pe întreg parcursul anului la majoritatea meseriilor, nu au putut fi eliminate decît în mică măsură prin optimizare cu ajutorul calculatorului.

2. Mai mult, încercarea de optimizare efectuată prelungind unele termene pînă la jumătatea anului viitor a scos în evidență menținerea în continuare a unor deficite de forță de muncă, datorate, pe de-o parte, normelor globale stabilite cu mare aproximație, iar pe de altă parte, unei supraîncărcări reale a planului în comparație cu posibilitățile concrete de lucru ale întreprinderii.

În consecință, se impun pe viitor : (a) revizuirea normelor existente pe baza experienței de pînă acum a întreprinderilor din cadrul Centralei și (b) elaborarea planurilor directoare anuale și bianuale pentru toate întreprinderile Centralei Industriale Navale cu ajutorul unor metode științifice, folosind instrumente de optimizare pe calculatorul electronic.



Metoda propusă pentru perfecționarea sistemului de conducere în sectoarele calde la Întreprinderea Metalurgică Aiud se bazează pe utilizarea sistemului SOFTPLAN în două etape : (a) programarea anuală, (b) programarea la nivel de lună, pe baza rezultatelor obținute la punctul a, cu alte cuvinte utilizarea pachetului de programe SOFTPLAN la două niveli diferite de detaliere și în mod succesiv. Schema de principiu (vezi fig. 18) prevede mai întîi încărcarea bazei de date folosind indicațiile din planul anual, corectarea erorilor și obținerea unor variante de plan anual optimizat. Apoi, pe baza variantei anuale adoptate, detalierea încărcărilor și capacitaților la nivel de lună și optimizarea încărcării posturilor pentru fiecare lună în parte.

Procesul la I.M.Aiud s-a desfășurat astfel :

1. S-au alcătuit următoarele 5 posturi pentru turnătorie : 1. modelărie, 2. sector neferoase, 3. sector piese diverse din fontă, 4. turnătorie de oțel I și 5. turnătorie de oțel II, iar în ceea ce privește forja, gruparea capacitaților s-a făcut pe utilaje de forjare și echipele care le deservesc astfel : 6. ciocan de 150 și 160 kgf, 7. ciocan de 400 kgf, 8. ciocan de 2000 kgf, 9. mașină de forjat orizontală de 160 tf și 10. presă de forjare liberă 3150 tf.

Pentru calculul capacitații medii disponibile la fiecare din aceste posturi s-au avut în vedere : ponderea ce o ocupă producția marfă planificată (care se livrează direct beneficiarilor externi) precum și manopera necesară executării pieselor pentru activitatea de întreținere și reparații din cadrul întreprinderii (consumul intern). Prin diferența față de total a acestor categorii de manoperă a rezultat capacitatea disponibilă pentru

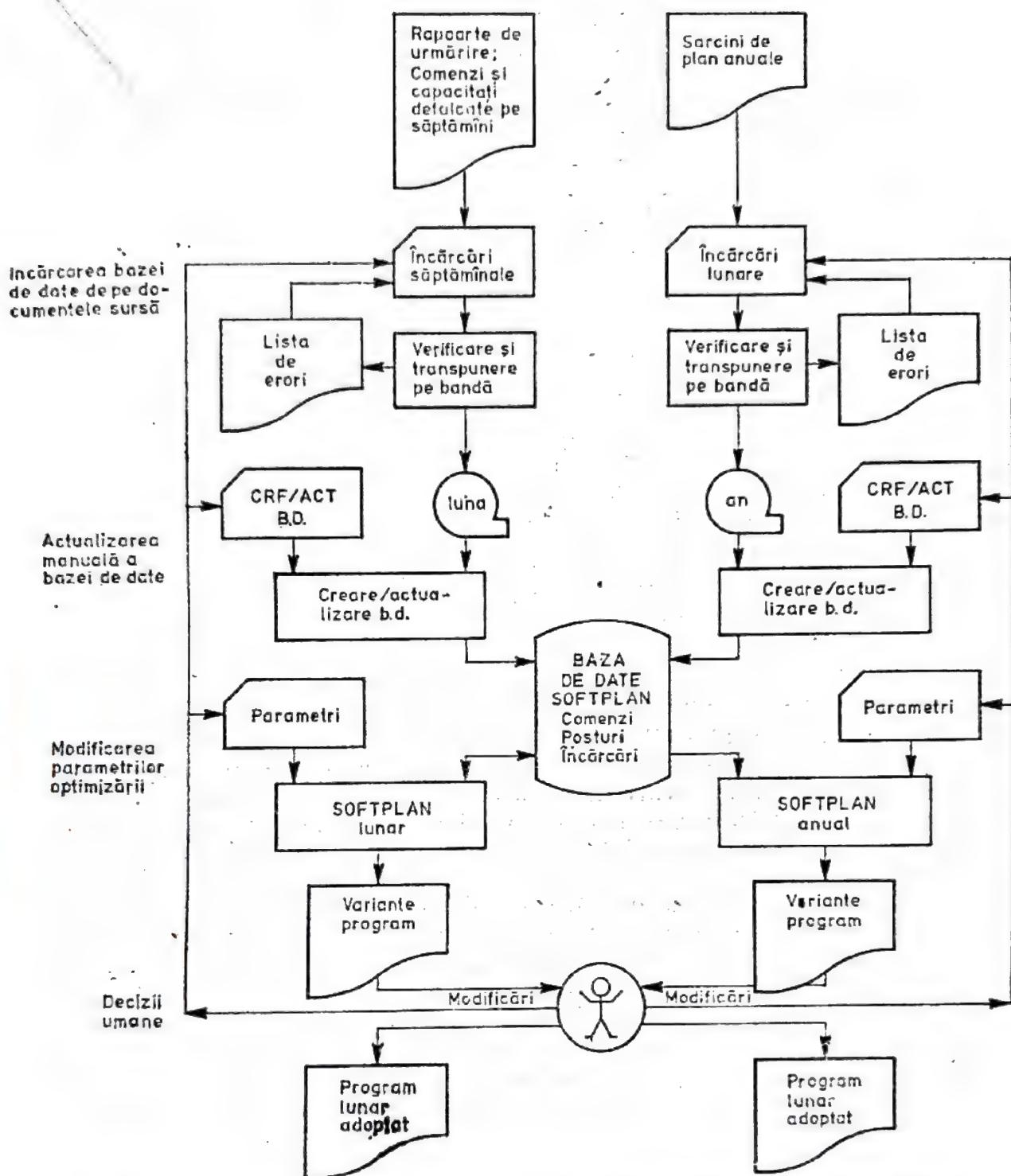


Fig. 18. — Programarea producției la două niveluri (anual și lunar) cu ajutorul sistemului SOFTPLAN (Intreprinderea metalurgică Aiud).

piesele ce merg la prelucrări mecanice și care interesează în mod deosebit întreprinderea din punctul de vedere al programării.

2. Pentru codificarea comenziilor au fost utilizate inscripțiile din comenziile beneficiarilor iar pentru produse s-au introdus 12 caractere având semnificații tehnologice. Au fost programate 16 comenzi, din care 12 comenzi pentru utilaj metalurgic și 4 comenzi din grupa de piese de schimb cu montaj.

3. Optimizarea încărcării posturilor de lucru a avut loc în două etape :

a. În prima etapă s-a realizat o programare pe întreg anul, avind ca perioadă de raportare luna. Timpul de început cel mai devreme (TID) și timpul de sfîrșit cel mai tirziu (TST) au fost stabiliți pentru fiecare comandă în funcție de termenul de livrare și răgazul necesar secției prelu-

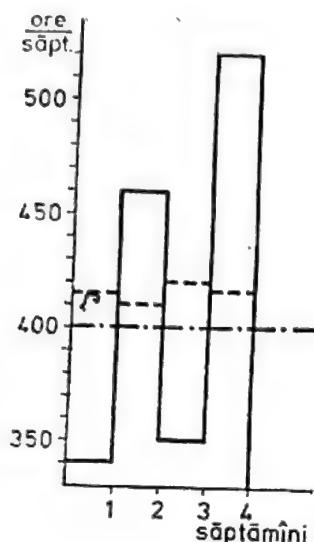


Fig. 19. — Încărcarea postului UPC09 înainte și după optimizarea la etapa a II-a, nivel lunar (Întreprinderea metalurgică Aiud) : - - - - capacitate ; —— încărcare înainte de optimizare. —— încărcare după optimizare.

crări mecanice pentru uzinare și montaj. Comenziile cu termene de livrare apropriate față de începutul anului au fost plasate înaintea celorlalte, ele fiind avantajate la programare și căpătind astfel o prioritate superioară.

Prin aplicarea succesivă a modulelor SOFTPLAN SIM1B și SIM2A s-a ajuns de la o supraîncărcare maximă inițială de 4 130 ore/lună (la postul CO105) la o supraîncărcare maximă în final de numai 455 ore/lună (pe postul UPC09), ceea ce reprezintă o reducere de aproape 10 ori a suprasarcinii inițiale.

b. În cea de-a două etapă, s-a realizat programarea săptămînală a comenziilor care trebuiau executate într-una din luni (pentru exemplificare am ales luna a 2-a). Capacitatea a fost diminuată corespunzător unei săptămîni, iar comenziile care se executau în luna respectivă au fost selectate și împărțite în subcomenzi operative (mai mici). Aplicînd din nou pachetul de programe „SOFTPLAN” s-a ajuns în cele din urmă la o distribuție aproape perfectă a comenziilor în intervalul de programare ales. (Postul UPC09, reprezentat în figura 19, ilustrează uniformizarea foarte eficace a sarcinii obținută în final cu acest pachet de programe).

*

Sistemul „SOFTPLAN” a fost utilizat cu succes și la programarea reparațiilor de motoare în cadrul Întreprinderii de Utilaj Greu pentru Construcții (IUGC) — București. În acest scop a fost necesară parcurgerea următoarelor etape :

a) Stabilirea posturilor de lucru (grupe de muncitori care execută aproximativ aceeași categorie de operații) în cadrul fiecărei linii de reparații de motoare. S-au stabilit următoarele 3 categorii de operații : (1) categoria operațiilor de demontare, spălare-degresare și constatare-triere ;

(2) categoria operațiilor de reparații linie arbore, mecanism de distribuție, mecanism bielă-manivelă, chiulasă, sistem de ungere, aparatură alimentare și anexe; (3) categoria operațiilor privind montajul general, rodajul și probele.

Succesiunea în timp a celor 3 categorii de operații este obligatorie. Pentru fiecare din cele 4 linii de reparații existente fiind folosiți muncitori specializați, care nu pot trece de la o linie la alta, iar în cadrul fiecărei linii efectuindu-se compartimentarea de mai sus, rezultă că pe total avem de a face cu 12 posturi de lucru, care au fost codificate 11, 12, 13, 21, 22 etc. (primul indice semnifică linia, iar al 2-lea categoria de operații care se execută în cadrul liniei).

b) *Cicloramele de reparații* reprezentând numărul de ore necesar efectuării fiecărei categorii de operații 1, 2 și 3 de mai sus, pentru fiecare tip de motor în parte, au fost calculate în felul următor: S-a pornit de la numărul total de ore-muncă necesar efectuării reparației fiecărui motor. Dintre acestea 10% s-au rezervat operațiilor pregătitoare (categoria 1), 70% reparațiilor subansamblelor (categoria 2) iar restul de 20% s-a considerat că se consumă cu montajul general, rodajul și probele (categoria 3).

În ceea ce privește calculul duratei fiecărei faze s-a avut în vedere că acestea sint în general diferite la motoarele mari (liniile 1, 2 și 4) față de cele mici (linia 3). Astfel, operațiile pregătitoare (categoria 1) durează în medie 2 zile la motoarele mari și 1 zi la cele mici, operațiile de reparații subansamble (categoria 2) durează cca. 8 zile la motoarele mari și numai 4 zile la cele mici ș.a.m.d.

S-a ales un interval de programare de o lună (30 zile), acesta fiind împărțit în 15 perioade programabile de cîte 2 zile fiecare:

c) *Capacitatea posturilor de lucru* 11, ..., 43 a fost calculată avind în vedere numărul muncitorilor disponibili pe fiecare linie precum și o repartizare (aproximativă) a acestora la fiecare categorie de operații sus menționată.

d) Pentru a stabili *volumul comenzilor de reparații* pe luna pentru care s-a efectuat programarea (aprilie) s-a pornit de la planificarea anuală a reparațiilor repartizată pe trimestre, din care s-a approximat o anumită cantitate pentru această lună. S-a avut în vedere programarea primită din partea SUGTC 3 Domnești-Voluntari, care apoi s-a centralizat pe tipuri de motoare. Au fost extrase din lista numerelor de inventar o parte din motoarele care au nevoie de reparații capitale și care urmează a fi programate în luna respectivă cu ajutorul sistemului „SOFTPLAN”.

La cele de mai sus, s-au adăugat pentru fiecare tip un număr de motoare respectind quantumul reparațiilor accidentale și care s-a approximat în funcție de nivelul reparațiilor din anul precedent. Au fost studiate 4 variante de încărcare, numărul reparațiilor accidentale crescind treptat de la o variantă la alta, pînă la un nivel considerat ca fiind maxim (vezi tabela 6).

Rezultatele obținute prin optimizarea încărcării posturilor cu sistemul „SOFTPLAN” au pus în evidență următoarele:

1. Reparațiile capitale planificate pentru luna aprilie la motoarele D105, D108, Tatra, Nobas, D110 și D103 (liniile 1, 2 și 3) se pot efectua integral, pentru acestea rezultind un program optim de intrare în reparație.

2. Reparațiile capitale prevăzute pentru luna aprilie la motoarele auto Saviem-Raba (linia 4) nu se pot efectua decit parțial cu capacitatele de reparații existente și ținînd seama de normele de reparații în vigoare.

Tabelă 6

CALCULUL COMENZILOR DE REPARAȚII ÎN VEDERE A PROGRAMĂRII CU AJUTORUL SISTEMULUI „SOFTPLAN” (4 VARIANTE)

Nr. crt.	Tip motor	Reparații capitale Luna aprilie			Anual (buc.)	Reparații accidentale Variante			
		Trim. II (buc)	(buc)	Serie motor		I	II	III	IV
1.	D 105	4	1	127682	110	6	7	8	9
2.	D 108	8	3	125927 125114 125555	50	2	3	3	4
3.	TATRA	3	1	3312253	30	—	1	2	3
4.	NOBAS	3	1	32611691	30	—	1	2	3
5.	D 103	2	1	329617	200	6	9	13	17
6.	D 110	5	2	326367 326713	30	2	2	3	3
7.	SAVIEM	14	4	31B5977 31B1063 31B1401 31B6685*)	36	—	—	2	3
8.	RABA	14	5	31B9193 31B9111 31B9557 31B8983*) 31B8216*)	36	—	—	1	3

*) Numai pentru variantele II, III și IV.

Pentru cele 3 motoare care se încadrează în capacitatea de reparație existentă s-a elaborat un program de intrare în reparație.

3. În vederea estimării disponibilului de capacitate pentru reparațiile accidentale s-au adăugat progresiv un număr de comenzi fictive ținind cont de datele statistice din anul precedent (vezi tabela 6).

Analizând gradul de acoperire al posturilor de lucru în fiecare din variantele avute în vedere se constată :

a) imposibilitatea efectuării unor reparații accidentale la linia 4-a (auto) unde chiar reparațiile capitale planificate nu pot fi satisfăcute decât în proporție de 30% cu actuala capacitate a liniei (vezi fig. 20);

b) posibilitatea efectuării a cel mult 1 reparație accidentală la linia 2-a în plus față de cele 2 reparații capitale planificate, varianta convenabilă fiind aceea situată între prima și a doua variantă analizată (vezi fig. 21);

c) posibilitatea efectuării suplimentare a cel mult 8 reparații accidentale față de cele 3 planificate la linia 3-a, varianta I fiind cea convenabilă (fig. 22);

d) posibilitatea efectuării tuturor reparațiilor accidentale estimate (13) la linia 1, varianta IV-a (maximă) fiind acceptabilă (vezi fig. 23).

4. Considerind totuși obligatorie realizarea variantei a IV-a (maximă) de reparații pentru perioada analizată, rezultă necesitatea suplimentării forței de muncă existente la reparațiile de motoare cu încă 25 muncitori repartizați după cum urmează :

- la linia 4 (auto), 13 muncitori;
- la linia 2, 7 muncitori;
- la linia 3 (motoare mici), 5 muncitori.

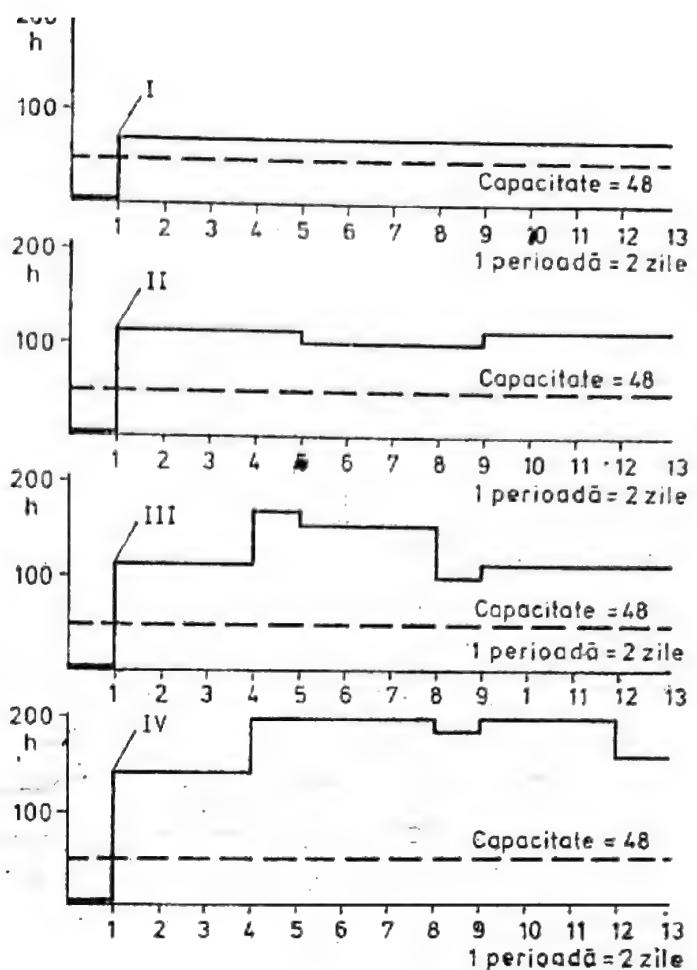


Fig. 20. — Postul de lucru 42 (IUGC) (I—IV: variante).

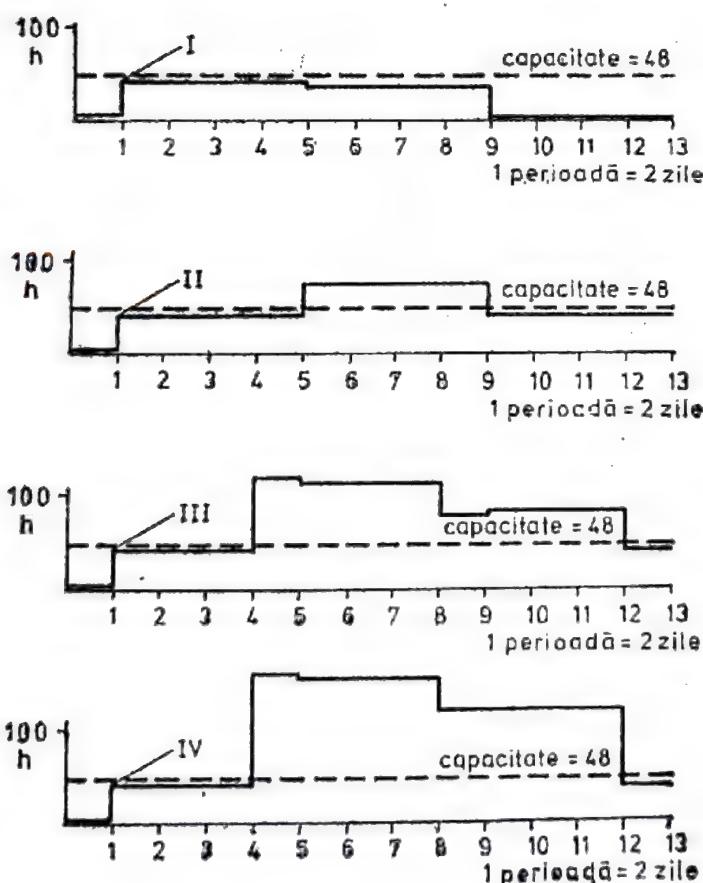


Fig. 21. — Postul de lucru 22 (IUGC) (I—IV: variante).

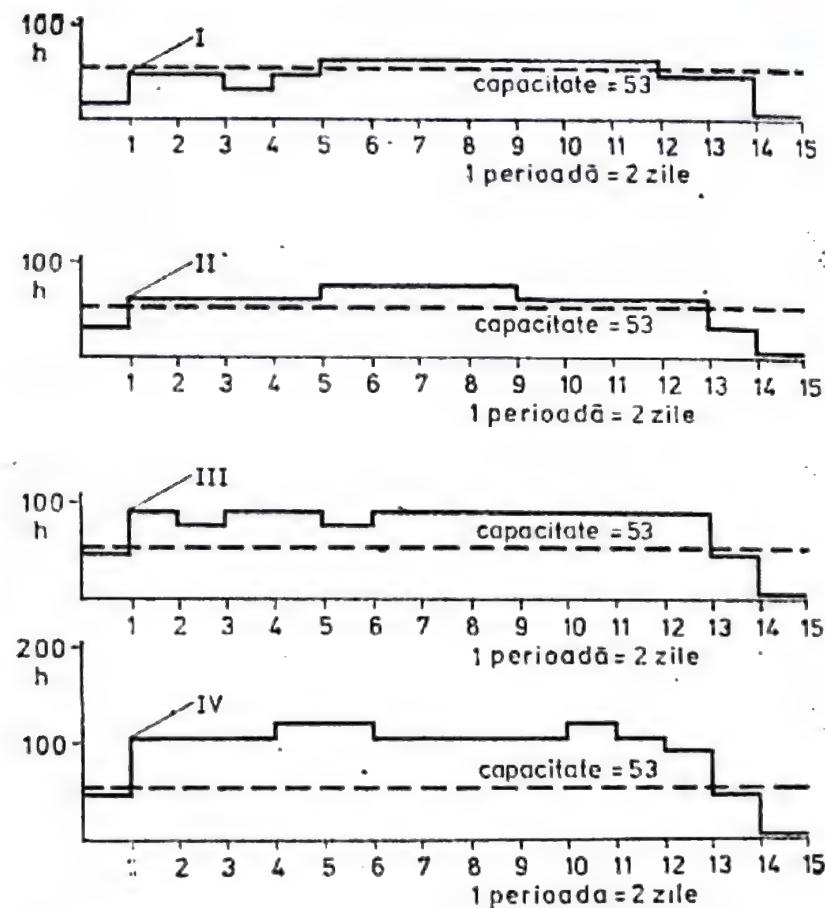


Fig. 22. — Postul de lucru 32 (I.U.G.C) (I—IV : variante).

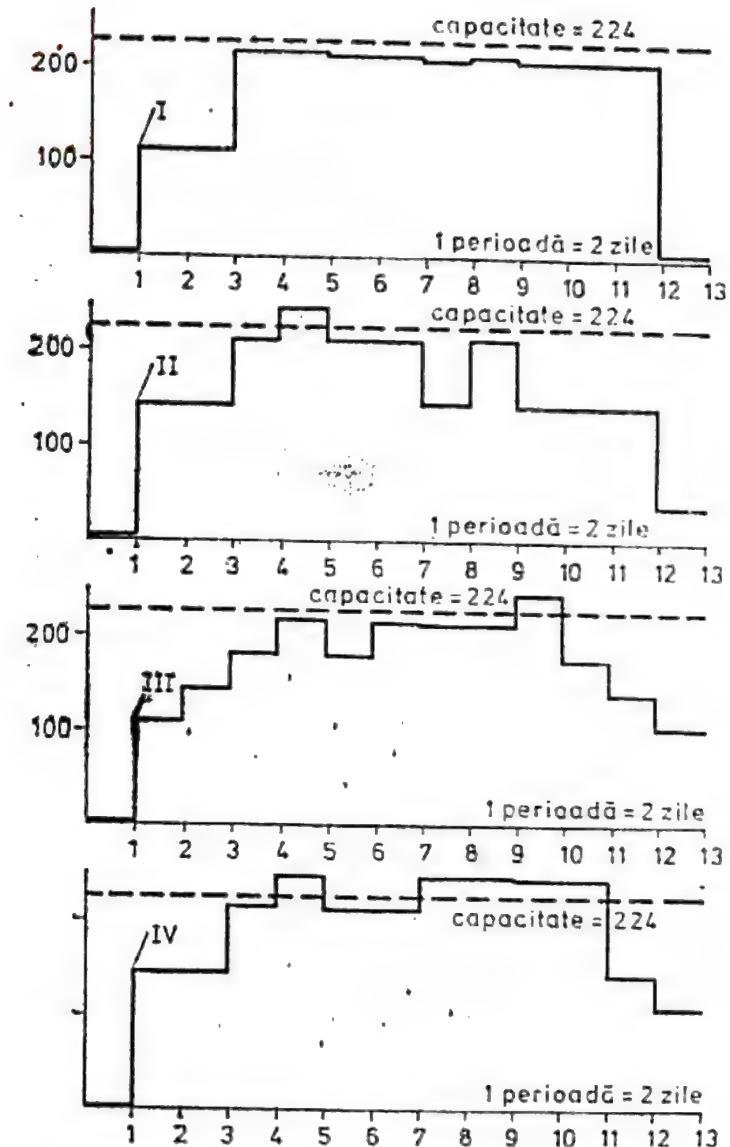


Fig. 23. — Postul de lucru 12 (I.U.G.C) (I—IV : variante)



5. În cazul în care suplimentarea forței de muncă nu este posibilă este necesar să se ia măsuri ca reparațiile de motoare ce depășesc limitele fiecarei linii arătate mai sus să fie dirijate din timp de către alte unități de reparații. În caz contrar, aglomerarea liniilor și depășirea capacitaților duce la nerespectarea normelor de reparații în vigoare și, implicit la realizarea unor reparații de slabă calitate, care pot duce în scurt timp la degradarea utilajului și scoaterea lui completă din funcțiune.

CONCLUZII

În urma experimentării sistemului „SOFTPLAN” în întreprinderile menționate, au rezultat o serie de observații în legătură cu aplicarea corectă a sistemului și mărirea eficacității lui potențiale în exploatare, precum și unele concluzii cu caracter general.

Referitor la observații, menționăm următoarele :

1) Pentru întreprinderile care au de asigurat o ritmicitate constantă a producției de-a lungul anului, legate fiind de planul de aprovizionare, import, colaborări etc., planul se poate defalca de la început pe trimestre sau luni (vezi I.E.M.I. sau Întreprinderea Electromureș), programarea de detaliu urmând să se facă cu ajutorul sistemului „SOFTPLAN” în interiorul trimestrului sau lunii respective.

2) Întreprinderile la care nu toate secțiile sunt inițial pregătite să preia sistemul „SOFTPLAN”, sau la care programarea apare necesară numai la unele secții (de exemplu, la I.M.Aiud, la sectoarele calde) pot utiliza sistemul, având grijă să prevadă pentru produsul finit o rezervă corespunzătoare în privința termenelor (la I.M.Aiud s-a prevăzut un decalaj egal cu ciclul de fabricație din sectorul prelucrări la rece).

3) Pentru întreprinderile în care baza o formează produsele cu ciclu lung de fabricație, este necesar să se prevadă în programul de fabricație și produse cu cicluri mai scurte și având o marjă de deplasare mai mare de-a lungul orizontului de plan (comenzi cu piese de schimb, subansamble, comenzi pentru terți etc.), oferind în acest fel calculatorului posibilități mai mari de optimizare.

4) Întreprinderile la care specificul producției îl formează lucrul în echipe, pe meserii (de exemplu, șantierele navale), pot utiliza sistemul „SOFTPLAN” pentru a obține uniformizarea manoperei pe meserii pornind de la graful execuției fiecarei produse desfășurat în timp, pe operații, și cumulind volumul de manoperă (vezi ICNPC Drobeta Turnu-Severin). Ciclogramele astfel calculate se introduc în baza de date „SOFTPLAN”, în urma optimizării rezultând o mai bună utilizare a potențialului echipelor de lucru, cărora li se asigură astfel un front de lucru continuu pe întreg anul. Eventualele vîrfuri de încărcare restante fiind cunoscute din vreme, există răgazul necesar de a fi diminuate sau chiar anihilate printr-o serie de măsuri organizatorice, ca de exemplu : deplasarea unor echipe de la o meserie la alta prin cursuri de policalificare și specializare, colaborări cu întreprinderi sau cu secții de același profil din cadrul întreprinderii, angajare temporară de personal, etc.

5) În sfîrșit, se poate remarcă faptul că aplicarea sistemului de programare „SOFTPLAN” pe perioade mai scurte, la nivelul unei secții cu puține posturi de lucru dar cu multe comenzi (vezi, de exemplu, Întreprinderea Electromureș), conduce la necesitatea eliminării comenziilor a căror manoperă pe post de lucru este mică în raport cu restul comenziilor.

Comenzile mici urmează să constituie rezerva secției, în cazul în care una din comenziile programate nu a putut fi incepută sau este temporar interrupță, ele având și rolul de elemente de uniformizare suplimentară aflate la dispoziția șefului de secție. Menționăm, de asemenea, că în cazul aplicării sistemului „SOFTPLAN” pe perioade mai mici (de exemplu, programarea producției pe săptămâni în cadrul unei luni sau în cadrul unui trimestru) și pe secții (de exemplu, la uzinaj și, separat, la montaj), este de dorit să se creeze un decalaj între secții, cu stocuri de piese calculate pentru ritmul mediu de montaj, în așa fel încât eventualele neregularități sau abateri de scurtă durată de la program să poată fi, în cea mai mare parte, compensate prin măsuri locale și să perturbe cît mai puțin programarea (vezi I.E.M.I. și Întreprinderea Electromureș). Metoda decalajului și rezervei minime de piese care să asigure continuitatea procesului este, de altfel, singura care poate fi aplicată atunci cînd uzinajul și montajul sunt organizate după metode diferite: fabricația de piese pe comandă-reper (eventual, cu tehnologie de grup) iar montajul pe comandă-produs – pentru beneficiarul extern (cazul secției cablaje a întreprinderii Electromureș).

În ceea ce privește concluziile cu caracter general, trebuie să remarcăm :

1. Fiecare din unitățile în care s-a experimentat metoda de programare și sistemul „SOFTPLAN” are anumite particularități, atât în ceea ce privește profilul și volumul producției cît și din punctul de vedere al organizării; De asemenea, există mari diferențe și la unii parametri tehnici, cum ar fi, spre exemplu, lungimea ciclurilor de fabricație a produselor, gradul de specializare a forței de muncă, nivelul tehnic de dotare etc. Pentru fiecare unitate, a fost găsită însă un domeniu de aplicabilitate a sistemului care a permis experimentarea pachetului de programe în bune condiții și obținerea unor rezultate satisfăcătoare.

2. Programarea cu ajutorul sistemului „SOFTPLAN” prezintă, printre altele, avantajul de a putea fi inițial efectuată pentru producția de bază a întreprinderii și având în vedere numai posturile de lucru așa zise „cheie”, urmînd ca programarea să fie extinsă ulterior, în funcție de nevoie, și la alte niveluri.

3. Acest gen de programare nu deranjează sistemul de conducere operativă deja existent în întreprindere (transmiterea sarcinilor, lansarea documentației, urmărirea realizărilor), necesitând numai informațiile strict necesare, care oricum trebuie să stăpînească în orice sistem de programare, fie el și manual.

4. Referitor la posibilitatea generalizării unor experiențe căpătate prin aplicarea produsului „SOFTPLAN” în întreprinderi și pe această bază *reconsiderarea sistemului și a metodei ca parte integrantă a unui model intelligent de conducere operativă a producției în întreprinderile constructoare de mașini*, menționăm că acest lucru este realizabil numai :

a) în condițiile extinderii aplicării sale la toate întreprinderile constructoare de mașini a căror producție este organizată pe locuri de muncă (posturi de lucru) ;

b) prin combinarea forței de calcul și a puterii de optimizare a modulelor SOFTPLAN, cu suplețea, adaptabilitatea și puterea de generalizare a unui model de conducere intelligent (M.I) de tip INTELEC.

Vom încerca să demonstrăm viabilitatea acestei din urmă afirmații în cadrul ultimului capitol al lucrării de față prin rezolvarea unor probleme concrete de conducere operativă a producției.

Conducerea operativă a producției cu ajutorul modelului „INTELEC”

Se poate conduce procesul de producție într-o întreprindere în mod intelligent cu ajutorul calculatorului electronic? Pentru a răspunde afirmativ la această întrebare a fost necesară precizarea termenului de *conducere operativă* precum și definiția modelului intelligent (M.I.) de *conducere operativă* (cap. 5). Modelul INTELEC ale cărui baze au fost prezentate în cadrul capitolului 3, ne-a servit drept instrument de realizare a unui model intelligent de *conducere operativă* a producției în ramura construcțiilor de mașini în care sarcina de planificare și programare a fost îndeplinită de sistemul SOFTPLAN (cap. 6). Utilizând *PROCEDURA* generală, valabilă pentru orice model de acțiune (cap. 4) și predicatele de particularizare [ESTE (CAZ PARTICULAR AL)] și detaliere [ESTE (îN ACEST CAZ)], (SE COMPUNE DÎN (ANSAMBLU)], [SE COMPUNE DIN (SUCCEZIUNE)] etc. se poate furniza practic orice informație referitoare la *conducerea operativă* a producției. Punerea în evidență a unor legături logice indirecte între două fraze prin intermediul unui șir de predicate, explorarea cadrelor relaționale ca și posibilitățile de generalizare a informațiilor oferite de modelul INTELEC adaugă valențe noi funcției de informare. În cadrul acestui capitol se prezintă și o posibilă dezvoltare în viitor a logicii de combinare a facilităților modelului INTELEC, cu instrumente de „forță” de calcul exprimând modele complexe de optimizare de tipul SOFTPLAN, idee care completează conceptul de model intelligent (M.I.) definit în cadrul acestei lucrări. Etapa superioară de integrare a calculatorului în procesul de producție, sub formă de calculator-robot intelligent, poate fi atinsă prin evoluția modelului prezentat — adăugarea unui analizor automat (A.A), completarea cu dispozitive speciale comandate de calculator etc. În ceea ce privește implementarea modelului INTELEC, se propune un sistem de gestiune a bazelor de date dotat cu un limbaj procedural, de tip SOCRATE, completat cu un limbaj funcțional realizat de autori (SBDSC). Un exemplu de creare și exploatare a structurii modelului INTELEC (vezi cap. 3) prin SBDSC demonstrează viabilitatea acestei soluții. Analizorul semantic al limbajului INTELEC poate fi realizat sub forma unor tabele de decizie. În ceea ce privește analizorul sintactic se propun mai multe soluții: limbajul de manipulare al bazei de date, module speciale scrise în limbajul de asamblare al calculatorului, limbi specializate pentru probleme de inteligență artificială de tip LISP. Referitor la ultima variantă este dat un exemplu concret de realizare a compilatorului pentru limbajul INTELEC folosind facilitățile oferite de LISP.



7.1. Obiective

În cadrul capitolului 5 am definit *conducerea operativă a producției* ca fiind un proces cibernetic compus din comandă, execuție, controlul execuției și decizia privind continuarea procesului. Acest ansamblu de acțiuni se referă la o serie întreagă de aspecte menționate (planificarea globală, programarea de ansamblu și programările curente), este diferit de la etapă la etapă și are anumite particularități în funcție de specificul producției, modul de organizare, gradul de automatizare etc. Am văzut că, pentru construcția de mașini, de exemplu, particularitățile se refereau la modul de pregătire a fabricației, la determinarea sistemului de producție și a formelor de organizare, la calculul anumitor parametri specifici precum și la mecanismul de ordonanțare. Pentru a stabili în fiecare moment *obiectivele* conducerii operative, a găsi căile de acțiune care să ducă în mod optim la atingerea obiectivelor, a comanda și controla eficacitatea acțiunilor, este nevoie, în general, de un sistem complicat și am văzut că în cadrul unui asemenea sistem *calculatorul electronic* poate juca un rol *activ*. Având în vedere rolul calculatorului în cadrul procesului de conducere operativă a producției am definit conceptul de *model intelligent (M.I.)* și am încercat să demonstrăm utilitatea lui referindu-ne la conducedrea producției în ramura construcțiilor de mașini. În cele ce urmează vom căuta să adincim unele aspecte ale conducedrii operative cu ajutorul modelului INTELEC – definit în cadrul capitolelor 3 și 4 – arătind că el este capabil să evolueze în direcțiile a, b și c (vezi cap. 5) de la simplu la complex, preluind funcțiuni ale gândirii și activității umane și, prin urmare îndeplinește toate condițiile pentru a fi considerat model intelligent în sensul acceptat de noi mai sus.

7.2. Calculatorul-consilier

Să presupunem că modelul INTELEC se află în funcțiune în întreprinderea care ne interesează. Că gen de întrebări îi putem adresa, la cerăspunsuri ne putem aștepta din partea calculatorului și cum putem folosi aceste răspunsuri pentru a conduce mai bine procesul de producție?

a. Atunci, cind adresăm o întrebare trebuie să avem în primul rînd în vedere că *a conduce operativ* înseamnă a acționa cu un *scop*, pentru atingerea unuia sau mai multor *obiective*, prin intermediul mai multor *operații* a căror *succesiune* este sau nu cunoscută *apriori*; Fiecare operație este (sau trebuie să fie) *definită*, are (sau trebuie să aibă) o *durată*, un *termen* de realizare, unul sau mai mulți *execuțanți*, un *responsabil* și se încadrează într-o serie întreagă de *restrictii* care condiționează realizarea ei în bune condiții și cu rezultatul scontat. De asemenea, pe lîngă gama rezultatelor posibile, pentru operațiile executate automat (cu calculatorul) trebuie precizați și *parametrii* instrumentului informatic (pachetului de programe). La sfîrșitul fiecărei operații se pune firesc întrebarea: am îndeplinit obiectivele urmărite? Dacă da, procesul se termină, dacă nu, atunci în funcție de rezultatul obținut (prin comparație cu cel posibil) se determină operația următoare.

Aceasta este de fapt **〈PROCEDURA〉** generală, valabilă pentru orice model de acțiune, înregistrată în modelul INTELEC (vezi cap. 4) și pe această structură au loc toate celelalte particularizări determinate de utilizarea predicatorului [ESTE (CAZ PARTICULAR AL)] sau detalieri realizate cu ajutorul predicatorilor [ESTE (IN ACEST CAZ)], [SE COMPUNE DIN (ANSAMBLU)], [SE COMPUNE DIN (SUCCESIUNE)] etc. Așadar ori de câte ori în baza de date nu există suficiente informații pentru a răspunde direct la întrebarea formulată de utilizator, sistemul poate apela la informațiile aparținind unui nivel de generalitate imediat superior, la limită recurgind chiar la **〈PROCEDURA〉** generală amintită. Invers, pentru a obține informații mai detaliate referitoare la un concept, noțiune, mod de lucru etc. vom căuta frazele explicative având predicatorile [ESTE (CAZ PARTICULAR AL)], [ESTE (IN ACEST CAZ)], [SE COMPUNE DIN (ANSAMBLU)], [SE COMPUNE DIN (SUCCESIUNE)] etc.

Să dăm un exemplu. Presupunem că avem la dispoziție modelul INTELEC în care se află înregistrată **〈PROCEDURA DE UTILIZARE SOFTPLAN〉** (cap. 3). Așa cum știm **〈PROCEDURA DE UTILIZARE SOFTPLAN〉** poate interveni cu succes atât în *programarea de ansamblu*, cât și în *programarea curentă* în cadrul conducerii operative a producției pentru ramura construcțiilor de mașini (vezi cap. 5). Acest proces, la rîndul lui, se înscrie în **〈PROCEDURA〉** generală despre care vorbeam. Cu alte cuvinte, este valabilă fraza :

〈CONDUCEREA OPERATIVA A PROducțIEI〉 [ESTE (CAZ PARTICULAR AL)]
〈PROCEDURA〉

și, pe baza considerațiilor făcute la capitolul 5, putem scrie
〈SCOP〉 [ESTE (IN ACEST CAZ)] **〈ELABORAREA SARCINILOR PE LOCURI DE MUNCA SI ESALONATE IN TIMP〉**

〈CONDUCEREA OPERATIVA A PROducțIEI〉 [SE COMPUNE DIN (ANSAMBLU)]
〈PROGRAMAREA DE ANSAMBLU〉 (SI) **〈PROGRAMAREA CURENTĂ〉** (PE) (I)
〈INTERVAL OPERATIV〉 (SI) **〈CONTROLUL REALIZARILOR〉** (PE) (I) **〈INTERVAL OPERATIV〉**

〈PRIMA OPERATIE〉 [ESTE (IN ACEST CAZ)] **〈PROGRAMAREA DE ANSAMBLU〉**

în care, (conform 3^º cap. 5) pentru **〈PROGRAMAREA DE ANSAMBLU〉** este valabilă următoarea **〈DEFINITIE〉** precum și **〈SELECTIA OPERATIEI URMATOARE〉**

〈PROGRAMAREA DE ANSAMBLU〉

〈PROGRAMAREA DE ANSAMBLU〉 [SE COMPUNE DIN (ANSAMBLU)] **〈DEFINITIE〉** (SI) **〈SELECTIA OPERATIEI URMATOARE〉**

〈DEFINITIE〉 [ESTE (IN ACEST CAZ)] **〈DEFINITIE AL/A PROGRAMAREA DE ANSAMBLU〉**

〈SELECTIA OPERATIEI URMATOARE〉 [ESTE (IN ACEST CAZ)] **〈SELECTIA OPERATIEI URMATOARE AL/A PROGRAMAREA DE ANSAMBLU〉**.

〈DEFINITIE AL/A PROGRAMAREA DE ANSAMBLU〉

〈SE〉 [ESALONEAZA IN TIMP FABRICATIA SI SE OPTIMIZEAZA INCARCAREA POSTURILOR DE LUCRU] (PE) **〈ORIZONT DE PLAN〉**

〈CONDITIE/RESTRICTIE〉 [ESTE (IN ACEST CAZ)] **〈REALIZARE〉** (AL/A)

(PL) **〈INDICATOR ECONOMIC DE PLAN〉**.

<SE> [ACTUALIZEAZA]* <PROGRAMAREA CURENTA> (PE) (1) <INTERVAL OPERATIV> (IN) <OPERATIA URMATOARE> (SI) <SE> <ACTUALIZEAZA>* <'CONTINUA PROGRAMUL.'> (IN) <MESAJ>

<PROGRAMAREA CURENTA> (PE) (1) <INTERVAL OPERATIV> are în vedere <CONTROLUL REALIZARILOR> (PE) (I-1) <INTERVAL OPERATIV>, cu alte cuvinte, pentru oricare interval operativ I, sint valabile următoarele definiții și condiții de selecție ale operației următoare :

<PROGRAMAREA CURENTA>

<PROGRAMAREA CURENTA> [SE COMPUNE DIN (ANSAMBLU)] <DEFINITIE> (SI) <SELECTIA OPERATIEI URMATOARE>
<DEFINITIE> [ESTE (IN ACEST CAZ) <DEFINITIE AL/A PROGRAMAREA CURENTA> <SELECTIA OPERATIEI URMATOARE> [ESTE (IN ACEST CAZ)] <SELECTIA OPERATIEI URMATOARE AL/A PROGRAMAREA CURENTA>

<DEFINITIE AL/A PROGRAMAREA CURENTA>

<SE> [DEFALCA] <PROGRAM DE FABRICATIE> (PE) (I) <INTERVAL OPERATIV>
<SE> [ADAUGA] <CONTROLUL REALIZARILOR> (AL/A) (I-1) <INTERVAL OPERATIV>
<SE> [ADAUGA] (NOU) <PL> <OBIECTIV> (AL A) (I) <INTERVAL OPERATIV>
<SE> [OPTIMIZEAZA INCARCAREA POSTURILOR DE LUCRU] (PE) (I) <INTERVAL OPERATIV>
<SELECTIA OPERATIEI URMATOARE AL/A PROGRAMAREA CURENTA>
<SE> [ACTUALIZEAZA]* <CONTROLUL REALIZARILOR> (PE) (I) <INTERVAL OPERATIV> (IN) <OPERATIA URMATOARE> (SI)
<SE> [ACTUALIZEAZA]* <CONTINUA PROGRAMUL> (IN) <MESAJ>

<CONTROLUL REALIZARILOR>

<CONTROLUL REALIZARILOR> [SE COMPUNE DIN (ANSAMBLU)] <DEFINITIE> (SI) <SELECTIA OPERATIEI URMATOARE>
<DEFINITIE> [ESTE (IN ACEST CAZ)] <DEFINITIE AL/A CONTROLUL REALIZARILOR>
<SELECTIA OPERATIEI URMATOARE> [ESTE (IN ACEST CAZ)] <SELECTIA OPERATIEI URMATOARE AL/A CONTROLUL REALIZARILOR>

<DEFINITIE AL/A CONTROLUL REALIZARILOR>

<SE> [INREGISTREAZA IN BAZA DE DATE] (PL) <PRODUS REALIZAT> (DIN) <PROGRAM DE FABRICATIE> (PE) (I) <INTERVAL OPERATIV>

<SELECTIA OPERATIEI URMATOARE AL/A CONTROLUL REALIZARILOR>

<DACA> (I) <INTERVAL OPERATIV> [ESTE MAI MIC DECIT]* (ULTIM) <INTERVAL OPERATIV>

⟨ATUNCI⟩ ⟨SE⟩ [ACTUALIZEAZA]* ⟨PROGRAMAREA CURENTA⟩ (PE) (I+1) ⟨INTERVAL OPERATIV⟩ (IN) ⟨OPERATIA URMATOARE⟩ (SI) ⟨SE⟩ [ACTUALIZEAZA]* ⟨'CONTINUA PROGRAMUL'⟩ (IN) ⟨MESAJ⟩

Pentru toate operațiile de mai sus este valabil

⟨CONTROLUL REALIZARII OBIECTIVELOR⟩

⟨DACA⟩ ⟨OPERATIE⟩ [ESTE EGAL CU] ⟨CONTROLUL REALIZARILOR⟩ (PE) (ULTIM) ⟨INTERVAL OPERATIV⟩
 ⟨ATUNCI⟩ ⟨SE⟩ [ACTUALIZEAZA]* ⟨'PROGRAM TERMINAT'⟩ (IN) ⟨MESAJ⟩

Cerind, în aceste condiții, de la consola terminalului de teletransmisie executarea ⟨PROGRAM PRINCIPAL⟩ (AVIND) ⟨PROCEDURA⟩ (EGALA CU) ⟨CONDUCEREA OPERATIVA A PRODUCERII⟩ se va obține următoarea secvență de instrucțiuni *):

P 1 : Se înlocuiește în ⟨SUBPROGRAM⟩ entitatea ⟨OPERATIE⟩ cu ⟨PROGRAMAREA DE ANSAMBLU⟩.

P 2 : Se trece la executarea ⟨SUBPROGRAM⟩.

S 1 : Se execută ⟨PROGRAMAREA DE ANSAMBLU⟩ conform definiției : se eșalonează fabricația și se optimizează încărcarea posturilor de lucru pe întregul orizont de plan.

S 2 : Se înregistrează operația (EXECUTAT) ⟨PROGRAMARE DE ANSAMBLU⟩.

S 3 : Se înregistrează rezultatele obținute la operația de mai sus.

S 4 : Se generează :
 ⟨MESAJ⟩ [ESTE (IN ACEST CAZ)] ⟨'CONTINUARE IMPOZIBILA'⟩.

S 5 : Se execută ⟨CONTROLUL REALIZARII OBIECTIVELOR⟩. Mesajul rămîne neschimbăt.

S 6 : Condiție neîndeplinită. Se trece la operația următoare.

S 7 : Se execută ⟨SELECTIA OPERATIEI URMATOARE⟩ a programării de ansamblu.
 ⟨OPERATIA URMATOARE⟩ [ESTE (IN ACEST CAZ)]
 ⟨PROGRAMARE CURENTA⟩ (PE) (1) ⟨INTERVAL OPERATIV⟩
 ⟨MESAJ⟩ ESTE (IN ACEST CAZ) ⟨'CONTINUA PROGRAMUL'⟩.

S 8 : Instrucțiune inoperantă în acest caz ; Se revine la ⟨PROGRAM PRINCIPAL⟩.

P 3 : Condiție neîndeplinită. Se trece la fraza următoare.

P 4 : Se înlocuiește ⟨OPERATIE⟩ din ⟨SUBPROGRAM⟩ cu ⟨OPERATIE URMATOARE⟩ stabilită la S 7. Urmează fraza 2 din ⟨PROGRAM PRINCIPAL⟩.

P 2 : Se execută ⟨SUBPROGRAM⟩.

S 1 : Se execută ⟨PROGRAMAREA CURENTA (PE) (1) ⟨INTERVAL OPERATIV⟩⟩.

S 2 : Se înregistrează operația (EXECUTAT) ⟨PROGRAMARE CURENTA⟩ (PE) (1) ⟨INTERVAL OPERATIV⟩.

S 3 : Se înregistrează rezultatele obținute.

S 4 : Se generează mesajul de continuare imposibilă.

S 5 : Se execută controlul realizării obiectivelor care însă nu modifică mesajul.

S 6 : Instrucțiune inoperantă. Se trece la fraza următoare.

*) S-au păstrat aceleasi notatii : P n - indică fraza n din ⟨PROGRAM PRINCIPAL⟩ iar S m - fraza m din ⟨SUBPROGRAM⟩ (vezi cap. 4).

S 7 : <SELECTIA OPERATIEI URMATOARE> prevede <CONTROLUL REALIZARILOR> pe același interval operativ. Se generează :
 <OPERATIA URMATOARE> [ESTE (IN ACEST CAZ) <CONTROLUL REALIZARILOR> (PE) (1) <INTERVAL OPERATIV>
 <MESAJ> ESTE (IN ACEST CAZ)] <'CONTINUA PROGRAMUL'>.

S 8 : Se revine la <PROGRAM PRINCIPAL>.

P 3 : Instrucțiune inoperantă : Se trece la fraza următoare

P 4 : Se înlocuiește <OPERATIE> din <SUBPROGRAM> cu <CONTROLUL REALIZARILOR> (PE) (1) <INTERVAL OPERATIV>.

P 2 : Se execută <SUBPROGRAM>.

S 1 : Se execută <CONTROLUL REALIZARILOR> (PE) (1) <INTERVAL OPERATIV>.

S 2 : Se înregistrează operația.

S 3 : Se înregistrează rezultatele.

S 7 : <OPERATIA URMATOARE> [ESTE (IN ACEST CAZ)] <PROGRAMAREA CURENTA> (PE) (2) <INTERVAL OPERATIV>.

etc.

Execuția continuă cu <CONTROLUL REALIZARILOR> (PE) (2) <INTERVAL OPERATIV>, apoi (PE) (3), (4), ș.a.m.d. pînă cînd se ajunge la (ULTIM) <INTERVAL OPERATIV> cînd după <CONTROLUL REALIZARILOR> se înregistrează mesajul <'PROGRAM TERMINAT'> care oprește procesul. <CONDUGEREA OPERATIVA A PRODUCTIEI> se desfășoară astfel după o procedură generală unitară, mărimea orizontului de programare și numărul intervalelor operative urmînd a fi precizate în funcție de datele problemei.

În cadrul <PROCEDURA> (vezi cap. 4) operațiile care se execută pot fi manuale sau automate ; Mai mult, fiecare operație poate fi la rîndul ei o procedură construită după același model, dintr-un ansamblu de operații manuale și automate executabile conform <PROGRAM PRINCIPAL> și <SUBPROGRAM>.

Această logică permite dezvoltarea din aproape în aproape a sistemului. Dacă, de exemplu, la momentul la care ne referim, nu există în baza de date alte informații referitoare la <PROGRAMAREA DE ANSAMBLU> și la <PROGRAMAREA CURENTA>, ele se execută manual, respectînd definițiile și restricțiile general valabile date mai sus. Dacă însă, într-o etapă ulterioară, se dorește generalizarea experienței căpătate, de exemplu, în ramura construcțiilor de mașini, atunci, pentru <PROGRAMAREA DE ANSAMBLU> se înlocuiește în <DEFINITIE AL/A PROGRAMAREA DE ANSAMBLU> fraza generală <SE> [ESALONEAZA IN TIMP FABRICATIA SI SE OPTIMIZEAZA INCARCAREA POSTURILOR DE LUORU] (PE) <ORIZONT DE PLAN> cu următorul set de instrucțiuni :

<PROGRAMAREA DE ANSAMBLU>

<DEFINITIE AL/A PROGRAMAREA DE ANSAMBLU>

<SE> [EXECUTA]* <PREGATIRE>¹⁾
 <SE> [EXECUTA]* <DEFINIREA SISTEMULUI DE PRODUCȚIE SI A FORMELOR DE ORGANIZARE>²⁾
 <SE> [EXECUTA]* <CALCULUL PARAMETRILOR DESFASURARII PRODUCȚIEI>³⁾

în care 1), 2) și 3) au semnificațiile menționate în cadrul cap. 5.

În faza de **〈PREGATIRE〉** se execută o serie de operații specifice, cum ar fi: alcătuirea specificației de producție, pregătirea documentației de lansare, defalcarea indicatorilor de plan și transformarea acestora în planul fizic al întreprinderii, verificarea realizării planului fizic și adoptarea programului anual. Mai departe, programul anual se împarte în subperioade, se alcătuiește lista de produse și se calculează necesarul de materii prime, materiale, piese, ansamble de fabricat, colaborările, importul etc., pe fiecare subperioadă, pregătind terenul, așa cum am văzut, pentru programarea curentă.

〈DEFINIREA SISTEMULUI DE PRODUCȚIE ȘI A FORMELOR DE ORGANIZARE〉 este strins legată de faza **〈CALCULUL PARAMETRILOR DESFASURARI PRODUCȚIEI〉** și determină în final, așa cum s-a arătat în capitolul 5, adoptarea formei de organizare celei mai potrivite pentru fiecare reper precum și calculul unor parametri generali ai desfășurării producției.

În mod analog, pentru **〈PROGRAMAREA CURENTA〉** (PE) (I) **〈INTERVAL OPERATIV〉** se poate adopta o detaliere asemănătoare cu aceea din **〈ORDONANTARE〉** (cap. 5).

〈PROGRAMAREA CURENTA〉

〈DEFINITIE AL/A PROGRAMAREA CURENTA〉

1. **〈SE〉** [ALEGE] **〈SUBPERIOADA〉**
〈SUBPERIOADA〉 [ESTE (IN ACEST CAZ)] **〈SUBMULTIPLU〉** (AL/A)
(I) **〈INTERVAL OPERATIV〉**.
2. **〈SE〉** [DEFALCA] **〈PROGRAM ANUAL〉** (IN) **〈SUBPROGRAM ANUAL〉**
(PE) (I) **〈INTERVAL OPERATIV〉**
〈SUBPROGRAM ANUAL〉 [ESTE (IN ACEST CAZ)] (PL) **〈COMANDA-PRODUS DE FABRICAT〉**
3. **〈SE〉** [DEFALCA PE SUBPERIOADE] **〈CAPACITATE〉** (AL/A) (PL)
〈POST DE LUCRU〉
4. **〈SE〉** [RECALCULEAZA PE SUBPERIOADE] **〈CICLOGRAMA DE FABRICATIE〉**
(AL/A) (PL) **〈PRODUS〉**
5. **〈SE〉** [INCARCA] (PL) **〈COMANDA-PRODUS DE FABRICAT〉** (AL/A)
〈SUBPROGRAM ANUAL〉 (PE) (I) **〈INTERVAL OPERATIV〉**
6. **〈SE〉** [ADAUGA (PL) **〈COMANDA-PRODUS IN CURS〉** (AL/A) **〈SUBPROGRAM ANUAL〉** (PE) (I-1) **〈INTERVAL OPERATIV〉** (SI) (PL) **〈COMANDA-PRODUS RESTANTA〉** (AL/A) **〈SUBPROGRAM ANUAL〉** (PE) (I-1)
〈INTERVAL OPERATIV〉 (SI) (PL) **〈COMANDA-PRODUS NOUA, PRIORITYARA〉** (AL/A) **〈SUBPROGRAM ANUAL〉** (PE) (I) **〈INTERVAL OPERATIV〉**
7. **〈SE〉** [SCOATE] (PL) **〈COMANDA-PRODUS AMINATA〉** (AL/A) **〈SUBPROGRAM ANUAL〉** (PE) (I) **〈INTERVAL OPERATIV〉**
8. **〈SE〉** [OPTIMIZEAZA INCARCAREA POSTURILOR DE LUCRU] (PE) (I)
〈INTERVAL OPERATIV〉.

Într-o fază ulterioară se poate trece, de exemplu, la automatizarea optimizării așezării în timp a fabricației produselor, atât pentru **〈PROGRAMAREA DE ANSAMBLU〉** cit și pentru **〈PROGRAMAREA CURENTA〉** pe fiecare interval operativ. În acest scop, așa cum am menționat în cadrul capitolului 6, se pot utiliza pachetele de programe cunoscute: CLASS, ORDO etc. În

cazul în care pentru optimizarea încărcării posturilor de lucru se utilizează pachetul de programe SOFTPLAN, procedura de 〈PREGATIRE〉 (vezi cap. 5) arată astfel :

〈PREGATIRE〉

1. 〈SE〉 [ALCATUIESTE] 〈SPECIFICATIE DE PRODUCTIE〉
2. 〈SE〉 [PREGĂTESTE] 〈DOCUMENTATIE DE LANSARE〉
3. 〈SE〉 [DEFALCA] (PL) 〈INDICATOR〉 (PE) (PL) 〈INTreprindere〉
4. 〈SE〉 [TRANSFORMA] (PL) 〈INDICATOR〉 (AL/A) 〈INTreprindere〉 (IN) 〈PLAN FIZIC〉 (AL/A) 〈INTreprindere〉
5. 〈SE〉 [EXECUTA]* 〈PROGRAM PRINCIPAL〉 (AVIND) 〈PROCEDURA〉 (EGALA CU) 〈PROCEDURA DE UTILIZARE SOFTPLAN〉
6. 〈DACA〉 〈REZULTAT〉 ESTE NESATISFACATOR 〈ATUNCI〉 〈SE〉 [MERGE LA]* (3) 〈FRAZA>*
7. 〈SE〉 [ADOPTA] 〈VARIANTA OPTIMA〉 (AL/A) 〈PLAN FIZIC〉 (AL/A) 〈INTreprindere〉
8. 〈VARIANTA OPTIMA〉 (AL/A) 〈PLAN FIZIC〉 (AL/A) 〈INTreprindere〉 [ESTE (IN ACEST CAZ)] 〈PROGRAM ANUAL〉
9. 〈SE〉 [IMPARTE] 〈PROGRAM ANUAL〉 (IN) (PL) 〈SUBPERIOADA〉
10. 〈SE〉 [ALCATUIESTE] 〈LISTA PRODUSE DE FABRICAT〉 (PENTRU) (FIECARE) 〈SUBPERIOADA〉
11. 〈SE〉 CALCULEAZA NECESAR MATERIALEJ (PENTRU) (FIECARE) 〈LISTA PRODUSE DE FABRICAT〉
12. 〈SE〉 [CALCULEAZA NECESAR COLABORATORI] (PENTRU) (FIECARE) 〈LISTA PRODUSE DE 〈FABRICAT〉

la care se adaugă :

〈DEFINIREA SISTEMULUI DE PRODUCTIE SI A FORMELOR DE ORGANIZARE〉

1. 〈SE〉 [EXECUTA] * 〈CALCULUL SISTEMULUI DE PRODUCTIE〉
2. 〈SE〉 [EXECUTA] * 〈ALEGEREA FORMEI DE ORGANIZARE〉 (PENTRU) (FIECARE) 〈REPER I〉
 〈CALCULUL PARAMETRILOR DESFASURARII PRODUCȚIEI〉
 〈SE〉 [EXECUTA] * 〈CALCULUL PARAMETRILOR〉 (PENTRU) (FIECARE) 〈REPER I〉.

În ceea ce privește 〈PROGRAMAREA CURENTA〉 (PE) (I) 〈INTERVAL OPERATIV〉, singura parte ce se modifică în programul menționat anterior este fraza 8 care devine :

〈PROGRAMAREA CURENTA〉

〈DEFINITIE AL/A PROGRAMAREA CURENTA〉

8. 〈SE〉 [EXECUTA]* 〈PROGRAM PRINCIPAL〉 (AVIND) 〈PROCEDURA〉 (EGALA CU) 〈PROCEDURA DE UTILIZARE SOFTPLAN〉

Este evident că, pentru a trece de la acest program general la conducederea efectivă a producției într-o întreprindere oarecare, să spunem în



⟨INTreprinderea x⟩, este necesară introducerea a încă unui nivel de detaliere, prin care să se precizeze :

— Care sunt indicatorii de plan referitori la ⟨INTreprinderea x⟩

⟨INDICATOR⟩ [ESTE COMPUS DIN (ANSAMBLU)] ⟨PRODUCTIE NETA⟩ (SI)
⟨PRODUCTIE MARFA⟩ (SI) ⟨PRODUCTIE MARFA LA EXPORT⟩ (SI)

⟨CHELTUIELI LA 1000 LEI FONDURI FIXE⟩ (SI)...

⟨PRODUCTIE NETA⟩ (AL/A) ⟨INTreprindere x⟩ [ESTE (IN ACEST CAZ)]
⟨VALOARE 1⟩

⟨PRODUCTIE MARFA⟩ (AL/A) ⟨INTreprindere x⟩ [ESTE (IN ACEST CAZ)]
⟨VALOARE 2⟩ etc.

— Care sunt produsele, structura, ciclogramele lor de fabricație, care este lista de comenzi-produs cu eventuale priorități, termene de început și/sau de sfîrșit ce constituie restricții de fabricație impuse de centrală, beneficiar intern sau partener al unui contract extern, care este capacitatea maximă de lucru grupată pe posturi de lucru și perioade ce poate fi utilizată în ⟨INTreprinderea x⟩ ș.a., acestea fiind *date de intrare* în pachetul de programe SOFTPLAN sau în oricare alt instrument de optimizare a încărcării dinamice a capacitaților de producție. Amintim că, pentru stabilirea listei inițiale de produse ce urmează a fi fabricate, pot fi folosite instrumente informatiche de armonizare globală între plan și resurse, de tip programare liniară cu una sau mai multe funcții economice obiectiv (vezi cap. 5). Lista inițială urmează să fie ajustată prin iterării succeseive care constă în : așezarea în timp a produselor în mod optim — revenire la calculul indicatorilor economici — comparație cu indicatorii propuși pentru ⟨INTreprinderea x⟩ — redistribuirea indicatorilor în cadrul centralei — reașezarea produselor în timp pentru a verifica posibilitatea fabricării lor în condițiile cerute și cu capacitatea disponibilă pentru ⟨INTreprinderea x⟩ — ș.a.m.d. pînă la obținerea unui ⟨PROGRAM ANUAL⟩ satisfăcător pentru ⟨INTreprinderea x⟩.

b. Tot în scopul unei prelucrări pe diverse nivele de generalitate se poate folosi și *metoda directă* de informare. În cadrul capitolului 3., paragraful b, punctul b 1. ne-am referit la selectarea informațiilor după anumite caracteristici („COLECTAREA SI VERIFICAREA DATELOR „SOFTPLAN” și „INTreprinderea x”)) și am indicat metoda căutărilor succeseive mergînd spre grade de generalitate din ce în ce mai mari (mai întii toate caracteristicile, apoi numai primele două și dacă și această combinație lipsește se listează procedura generală de ⟨COLECTARE SI VERIFICARE A DATELOR⟩). Nu insistăm mai mult asupra acestei metode larg utilizată astăzi în marile baze de date pentru informarea selectivă pe baza descoperirii cuvintelor „cheie” atașate diverselor texte.

c. Față de sistemele de informare obișnuite, modelul INTELEC mai prezintă o facilitate, și anume, aceea de a putea face apel la rutina de informare pe parcursul executării unor programe, în aşa fel încit utilizatorul să poată decide singur, sau cu ajutorul unor proceduri automatizate pe calculator, cea mai bună continuare într-o situație de impas. Să dăm un exemplu.

Presupunem că avem la dispoziție modelul INTELEC în care se află înregistrată ⟨PROCEDURA DE UTILIZARE SOFTPLAN⟩ (cap. 3). Presupunem, de asemenea, că ne aflăm la fază 4 ⟨PROCEDURA DE OPTIMIZARE SOFTPLAN⟩ care se execută conform ⟨PROGRAM PRINCIPAL⟩ și ⟨SUBPROGRAM⟩ (cap. 4). Am văzut că dacă pe parcursul execuției lipsesc informațiile necesare con-

tinuării procesului, apare mesajul 'CONTINUARE IMPOZIBILĂ'. În cazul în care dispunem însă de rutina <SALVARE>, la consola terminalului vor apărea o serie de informații pe baza cărora utilizatorul poate decide continuarea procesului (informațiile se refereau la <REZULTAT> obținut pînă în acel moment, la <PARAMETRU> al instrumentului informatic utilizat, la <CONDITIE/RESTRICTIE> amintind condițiile în care a fost executată operația precedentă etc.). La un nivel de inteligență superior, concretizat prin rutina <DECIZIE>, toate operațiile cu ajutorul cărora utilizatorul evaluează situația și decide o continuare posibilă se automatizează. În acest din urmă caz există însă, așa cum am arătat, riscul unei decizii în necunoștință tuturor condițiilor particulare în care are loc procesul, a aprecierii în virtutea experienței trecute consemnate în baza de date și în funcție de gradul de similitudine cu situația nou creată, ceea ce, mai ales atunci cînd experiența nu este suficient de bogată, duce la lungirea procesului și adoptarea în final a unei soluții mai puțin satisfăcătoare. *Eficiența optimă a modelului se atinge, așadar, prin folosirea combinată a cazurilor de decizie manuală și decizie automată*, cele dintii contribuind la îmbogățirea tezaurului de informații înscrise în baza de date, cele din urmă servind la obținerea directă a soluțiilor de către utilizatorii mai puțin inițiați.

d. Tot în legătură cu consultarea calculatorului în procesul de conducere operativă a producției, amintim aici posibilitatea folosirii modelului INTELEC, în mod conversațional, pentru a obține unele informații specifice acestui model.

Așa cum am arătat în cadrul capitolului 3, există mai multe procedee de a obține aceste informații :

- cu ajutorul predicatelor [ESTE (CAZ PARTICULAR AL)] și [ESTE (IN ACEST CAZ)];
- prin prelucrarea succesivă, la nivele de detaliere din ce în ce mai adînci, a unor cadre relaționale. Pentru aceasta este nevoie de algoritmi de prelucrare speciali (vezi de exemplu algoritmul dat în capitolul 3, paragraful b.3.);
- prin punerea în evidență a unor legături logice indirecte (descoperirea unui sir de predicate) între două sau mai multe informații existente în baza de date.

Mentionăm că, atât în cazul explorării cadrelor relaționale cît și în acela al găsirii legăturilor indirecte între două sau mai multe noțiuni, durata de căutare crește exponențial cu numărul nivelelor explorate, parcurgerea făcîndu-se pe arbori cu un mare număr de ramuri. Chiar dacă se folosesc uneori tehnici speciale de căutare, ca de exemplu : căutarea după optim (mini/max), strategia „ramifică și înlănuie” (branch and bound), tehnica „alfa-beta”, §.a. ([73] cap. 4, p. 93—132), durata de căutare poate constituî o piedică în obținerea unui răspuns în timp util. De aceea, în alcătuirea programelor de acest tip se va avea în vedere introducerea unei limite de timp pentru obținerea rezultatului, depășirea acesteia fiind anunțată la consolă pentru avertizarea utilizatorului.

De asemenea, pe măsură ce intinderea domeniului explorat se mărește și în cadrul legăturilor intervin noțiuni din ce în ce mai generale, apar tot mai multe asocieri nesemnificative de fraze, sau chiar asocieri greșite din punct de vedere tematic, și aceasta deoarece asocierile nu țin seama, în general, de contextul în care frazele rămîn valabile, de valoarea

de adevăr a frazei raportată la condițiile de înregistrare. Dacă însă în baza de date relaționale (BDR) atașăm fiecărei fraze un indice (PF) care să exprime probabilitatea de adevăr a frazei respective (vezi capitolul 3), prin înmulțirea acestor probabilități vom ajunge la indicele cu care poate fi creditată informația descoperită prin lanțul de predicate ce leagă noțiunile de la care s-a pornit. Acest indice poate fi și el afișat la consolă odată cu prezentarea rezultatului.

De exemplu, fie textul următor :

1. <CONDUCEREA OPERATIVA> (AL/A) <PRODUCTIE> [ESTE (CAZ PARTICULAR AL)] <PROCES> (DE) <CONDUCERE>
2. <PROCES> (DE) <CONDUCERE> [SE COMPUNE DIN (SUCCESIUNE)] (PL) <OPERATIE>
3. <OPERATIE> [ESTE (ÎN ACEST CAZ)] <STRINGERE> (AL/A) <RECOLTA>
4. <STRINGERE> (AL/A) <RECOLTA> [FACE PARTE DIN] <CAMPANIE> (DE) <TOAMNA>
5. <TOAMNA> [INFLORESTE] (PL) <CRIZANTEMA>

Dacă ne referim la conducerea operativă a producției din întreprinderi și ne limităm la primele două fraze, lucrurile sint în ordine.

De îndată, însă, ce extindem domeniul și căutăm și alte legături, apare fraza 3 legată de fraza 2 prin entitatea <OPERATIE> dar având aici o particularizare, <STRINGERE> (AL/A) <RECOLTA>, care ieșe din sfera noastră de interes. Dacă se continuă, apare și fraza 4 care se depărtează și mai mult de obiectul investigației noastre, ca și în cele din urmă fraza 5, legată prin entitatea <TOAMNA> de fraza precedentă, să sugereze o apropiere aproape paradoxală între conducerea operativă a producției și crizanteme !

Dacă avem grija, însă, ca în algoritmul de căutare să verificăm, de fiecare dată, apartenența frazelor găsite la domeniul care ne interesează sau la domenii înrudite, rezultatele obținute vor fi, în general, corecte.

Fie textul :

1. <CONDUCEREA OPERATIVA> (AL/A) <PRODUCTIE> [ESTE (CAZ PARTICULAR AL)] <PROCES> (DE) <CONDUCERE>
2. <PROCES> (DE) <CONDUCERE> [SE COMPUNE DIN (ANSAMBLU)] <VERIFICARE> (SI) <CONTROL>

Din asocierea ultimelor două fraze ar rezulta că și conducerea operativă a producției se compune exclusiv din verificare și control, ceea ce este adevărat numai în parte.

Dacă, însă, vom ține seama că fraza 2 nu exprimă decât un caz particular, care acoperă, să zicem, 10% din totalul proceselor de conducere avute în vedere, rezultă că și concluzia noastră referitoare la conducerea operativă a producției nu poate fi creditată decât cu un coefficient de valabilitate de maximum 10%.

e. În sfîrșit, trebuie să menționăm – mai mult cu titlu de experiment – posibilitatea modelului INTELEO (vezi capitolul 3, paragraful b.5) de a furniza și unele informații *sintetice* pe baza generalizării cazurilor particulare găsite în baza de date la momentul formulării întrebării. Această facilitate trebuie utilizată de asemenea cu prudență deoarece aşa cum am văzut în exemplul anterior, apropierea unor noțiuni din domenii foarte diferite riscă să creeze confuzii. Dacă domeniul de referință este însă suficient de restrins (ca în cazul exemplului din paragraful b.5 capitolul 3) se obțin, în general, rezultate interesante.

7.3. Calculatorul-executant

În cadrul capitolului 5, cu ocazia prezentării conceptului de conducere operativă a producției aplicat întreprinderilor constructoare de mașini, au fost date o serie de proceduri de calcul («CALCULUL SISTEMULUI DE PRODUCTIE», «CALCULUL FONDULUI NOMINAL DE TEMP F», «CALCULUL PARAMETRILOR» (PENTRU) (FIECARE) (REPER 1) și.a.) și proceduri complexe de optimizare («PROCEDURA DE OPTIMIZARE SOFTPLAN») în care calculatorul avea în principal rolul de *executant* al unor serii de calcule, comparații logice și reveniri (iterații) având de atins obiective precise formulate *a priori*. Volumul calculelor și al datelor vehiculate era mare în comparație cu informațiile sintetice prezentate, atât pe parcurs, pentru control, cit și la sfîrșit, drept rezultat final, utilizatorului. Acesta din urmă avea posibilitatea (vezi de exemplu «PROCEDURA DE OPTIMIZARE SOFTPLAN») să ceară noi variante în alte «CONDITII/RESTRICTII» sau cu alții «PARAMETRI», să compare rezultatele și să adopte sau nu una din soluții. De asemenea, dacă consideră necesar, putea cere calculatorului informații suplimentare referitoare la rezultatele obținute în alte întreprinderi cu același pachet de programe sau cu altele similare, la «CONDITIILE/RESTRICTII» și «PARAMETRII» cu care s-a lucrat acolo etc.

Utilizarea combinată, în cadrul același sistem interactiv om-calculator, pe de o parte a unor programe de calcul și optimizare bazate pe modele complexe iar pe de altă parte a unor proceduri de informare care să servească la alegerea parametrilor necesari executării programelor sus menționate, constituie, așa cum am arătat în capitolul 5, o *etapă superioară* de integrare a calculatorului în cadrul modelului intelligent de conducere operativă a producției (M.I.). Programele de calcul (vezi de exemplu modulele de optimizare SOFTPLAN, SIM1B și SIM2A – cap. 6) reprezintă „forța” calculatorului bazată pe posibilitatea acestuia din urmă de a efectua cu mare repeziciune și precizie calcule aritmetice și logice complicate în cadrul unor *situări formale bine definite*. Dar, pentru a defini aceste situații formale, atunci cînd se dispune de un set redus de specificații mai mult sau mai puțin legate de problema de rezolvat, este necesară o *consultare intelligentă* a bazei de cunoștințe de care dispunem, după strategii automate sau manuale, consultare realizată în cazul nostru cu ajutorul modelului INTELEC. În figura 24 este reprezentat acest cuplu „inteligentă-forță” pentru aplicarea pachetului de programe SOFTPLAN în fazele «PROGRAMARE DE ANSAMBLU» și «PROGRAMARE CURENTA» ale procesului de conducere operativă a producției în întreprinderile constructoare de mașini.

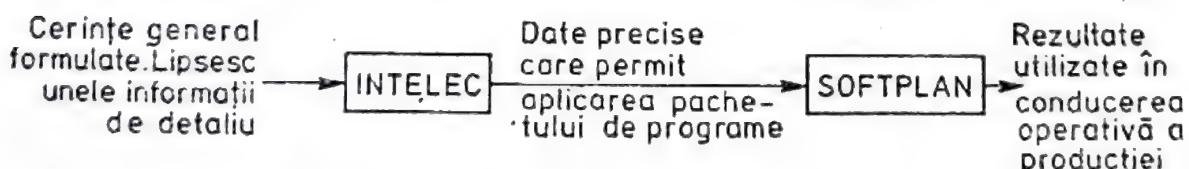


Fig. 24. – Cuplul „inteligentă-forță” în cazul aplicării pachetului de programe SOFTPLAN pentru conducerea operativă a producției în ramura construcțiilor de mașini.

O serie de obiective globale (satisfacerea indicatorilor de plan, îndeplinirea cerințelor formulate de beneficiari, obținerea unei înalte ritmicități și continuități a producției etc), se transformă prin utilizarea modelului INTELEC în *obiective precise* (datele de începere și de terminare a fabri-



cație produselor, mărimea optimă a loturilor, obținerea unui nivel minim de supraîncărcări și goluri de producție pe toată perioada programată, etc.), care se realizează în *condiții precise de lucru* prin furnizarea datelor de intrare necesare funcționării pachetului de programe SOFTPLAN (lista comenziilor—produs programabile, numărul de produse pe comandă, restricțiile în privința începerii și terminării fabricației pentru fiecare comandă în parte, cicloramele de fabricație ale produselor, capacitatele limită ale posturilor de lucru etc.).

Gradul de automatizare al diverselor strategii determină gradul de aplicare intelligentă a calculatorului în faza de consultare. După cum am văzut, modelul INTELEC prezintă unele facilități de *căutare, asociere și grupare* a informațiilor în baza de cunoștințe deși, deocamdată, în această primă versiune, *combinarea* lor în scopul rezolvării unei anumite probleme este redusă. Principiile care stau la baza modelului INTELEC, expuse în cadrul capitolului 3 al prezentei lucrări, ne fac să intrevedem însă posibilitatea dezvoltării în viitor a logicii de combinare a facilităților în scopul extinderii zonei de automatizare în paralel cu mărirea gradului de generalitate al informării. În acest fel modelul va răspunde din ce în ce mai bine necesității de informare în cazul unor situații reale foarte diferite ca aspect, canalizind răspunsurile utilizatorului către informațiile precise cerute de modulele de optimizare ce intervin în diverse etape ale conducerii operative a producției.

Asemenea modele, realizate astăzi în lume pentru alte domenii de activitate și menționate în literatura de specialitate *) ne fac să sperăm în atingerea în viitorii ani și a acestui obiectiv.

7.4. Calculatorul-robot

Amintim aici definiția dată în cadrul cap. 5 pentru această noțiune și care se referă exclusiv la modelul de conducere operativă a producției prezentat, denumit de noi *model intelligent (M.I.)*. Spuneam atunci că termenul de *calculator-robot* este asociat de noi sistemelor complexe automate care își asumă concomitent rolul de *decident și de executant, prelucrind permanent informații provenite din mediul înconjurător și acționând totodată asupra acestuia*. Aceste intrări/ieșiri de informații pot fi culese, respectiv lăvrate, *direct* de calculator, prin intermediul unor instalații de automatizare (convertori analog/digital și digital/analog; detectori de semnale,

*) Ira Goldstein și Bruce Roberts, *Using frames in scheduling in Artificial Intelligence: An MIT Perspective*, edited by Patrick H. Winston and Richard H. Brown, MIT Press, Cambridge, Massachusetts London — England, 1979. Se menționează experimentul de cuplare al programului *NUDGE* care acceptă cerințe informale și produce un calendar conținând conflicte posibile precum și o mulțime de strategii asociate pentru rezolvarea acestor conflicte, cu programul *BARGAIN* care rezolvă aceste conflicte prin tehnici de analiză a deciziei tradiționale. *NUDGE* utilizează o bază largă de cunoștințe pentru a putea sesiza și coriga cererile de escalonare informale, pentru a putea suplini detaliile care lipsesc, pentru a rezolva inconsistențele și a determina opțiunile potrivite în cadrul unui plan care conține rezultatele scontate. Un limbaj de reprezentare simplificat, *FRL-0*, construit special în acest scop, permite ca proprietățile din baza de cunoștințe să fie descrise prin comentarii, abstracții, defecți, referiri indirekte la alte proprietăți etc. Dar fiind faptul că manipularea unor asemenea limbaje nu este astăzi încă posibilă la punct, potențialul de interacționare și interfață a noțiunilor în baza de cunoștințe nefiind încă suficient cunoscut, rezultatele obținute cu acest model simplificat nu sunt deocamdată concluziente.



impulsuri, unde, temperaturi, presiuni etc. și, respectiv, emițători ai acestora) sau, pot fi numai *semnalate* omului, prin afișarea, la consola terminalului, a unei *cereri de informație* sau a unei *informații-rezultat*. În același mod, acțiunea asupra mediului poate fi *directă*, prin comanda unor dispozitive de acționare electrice, pneumatice etc., sau, tot *prin intermediul omului*, calculatorul descriind acțiunea pe care omul trebuie să o execute. Asupra modului de acționare direct nu vom insista aici el constituind obiectul de realizare și perfecționare al așa numitelor *calculatoare de proces*, întâlnite cel mai des astăzi în conducerea proceselor de producție cu flux continuu și care se desfășoară în condiții grele (mediu toxic, temperatură ridicată etc.) Ne vom opri puțin însă asupra celui de-al doilea tip de conducere complexă, indirectă, menționat mai sus.

Să presupunem că în schema de utilizare intelligentă a forței de calcul din figura 24 introducem următoarea modificare: rezultatele obținute prin aplicarea pachetului de programe SOFTPLAN nu mai sunt comunicate omului — care avea posibilitatea să le accepte sau să încerce noi variante de aplicare — ci unui analizor automat (AA) construit tot cu ajutorul limbajului INTELEC și, ca atare, făcind parte integrantă din modelul cu același nume (vezi figura 25). Pe baza unor criterii de selecție precis formulate și având posibilitatea de eliminare a variantelor neconvingătoare prin compararea cu cazurile similare întâlnite în aceeași întreprindere sau în altele asemănătoare, analizorul automat ar putea prelua funcția de selecție îndeplinită până atunci de decidentul uman, repetind atâtea variante (cerind în acest scop, la nevoie, și alte informații suplimentare în vederea luării unei decizii juste de continuare) până cînd ar obține o *situație satisfăcătoare direct aplicabilă în producție*. Este ușor de văzut că în acest caz ar fi necesară *declanșarea automată a optimizării* diverselor variante propuse de modelul INTELEC, fără a mai aștepta intervenția omului, acesta ne mai fiind la curent cu numărul și forma încercărilor efectuate pe parcurs de calculator și primind numai rezultatul final, direct aplicabil.

Pentru a trece la o etapă superioară de integrare a calculatorului în procesul de producție ar fi necesare *dispozitive speciale comandate de calculator* care, pe baza rezultatului final obținut să realizeze: lansarea automată a documentației de fabricație a produsului sub formă scrisă sau pe memorie magnetică; informarea, la locul de muncă, la cerere, prin dispozitive automate de afișare de tip „display” a ceea ce trebuie executat conform programului adoptat; raportarea, de la locul de muncă, din partea personalului de control, prin intermediul acelorași dispozitive automate de teletransmisie, a rezultatelor obținute, a abaterilor de la program precum și a indisponibilității utilajelor. În cazul în care mașinile-unelte sunt automate, raportarea se poate face direct de către mașină.



Fig. 25. — Modelul complex prevăzut cu analizor automat al rezultatelor.

Modele ale procesului de producție, (de exemplu INTELEC-2, vezi fig. 26) realizate pe calculator ar putea prelucra în permanență anomaliiile,

abaterile voite sau nevoite de la program (prin intermediul INTELEC-1, vezi fig. 26) construind soluția optimă și revizuind în consecință programul pe fiecare loc de muncă. Se înțelege însă că o asemenea soluție este foarte costisitoare, nu numai din punctul de vedere al echipamentului electronic de calcul și de automatizare care trebuie să funcționeze continuu, cu mare fiabilitate, viteză și precizie ridicată, volum mare de memorie

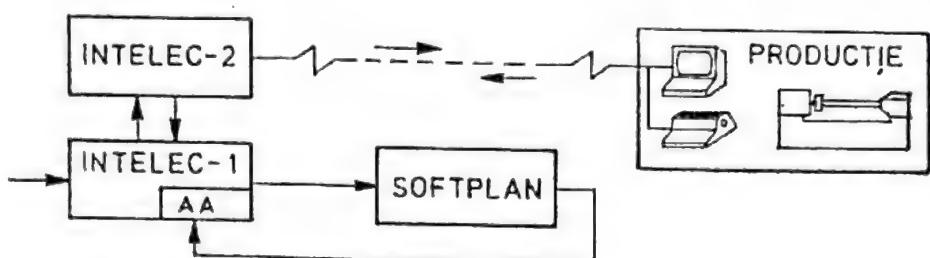


Fig. 26. – Conducerea directă prin calculator a producției.

etc., dar și din punctul de vedere al *efortului de realizare al modelelor procesului de producție care să reflecte fidel situația întreprinderii în cauză*, știut fiind că particularitățile unei întreprinderi sunt extrem de numeroase iar modelul trebuie să fie suficient de exact pentru a da rezultatele scontate.

Sintem de părere că limitarea, deocamdată, la schemele mai simple prezentate anterior ar corespunde situații pe o poziție realistă în ceea ce privește posibilitățile conducerii operative a producției în țara noastră.

7.5. Realizarea modelului INTELEC

În cadrul capitolului 3 a fost definit limbajul INTELEC și baza de date relațională (*BDR*) ca suport informațional al acestui limbaj.

Pentru implementarea *BDR* este necesar un instrument informatic capabil să asigure gestionarea completă și regăsirea rapidă a informațiilor și care să permită totodată realizarea oricărei structuri de date. De asemenea, trebuie asigurat schimbul de mesaje, prin terminale amplasate la distanță față de centrul de calcul precum și posibilitatea consultării simultane a *BDR* de către mai mulți utilizatori.

Implementarea *predicatelor interne* ale limbajului INTELEC (de tip [EXECUTA]*, [INREGISTREAZA] *, [INLOCUIESTE] * etc. vezi tabela 1, capitolul 3) presupune posibilitatea construirii unor proceduri specifice de căutare, regăsire, memorare temporară, calcule aritmetice și logice și afișare a rezultatelor — operații care reclamă existența unui *limbaj procedural complet al instrumentului informatic* pe baza căruia este realizată *BDR*.

Pentru calculatoarele FELIX-C512 poate fi folosit, ținând seama de cerințele de mai sus, sistemul de gestiune a bazelor de date SOCRATE, dotat cu limbaj procedural propriu, cu macrogenerator și editor de programe, cu sistem conversațional independent MULTITASKING permitând gestionarea mai multor linii de terminale de teletransmisie și având la bază un procedeu performant de regăsire a informației care utilizează principiul memoriei virtuale.

Limbajul **SOCRATE** a fost recent, completat cu o serie de rutine care permit calcule cu numere zecimale, imprimarea unor siruri cu lungimi mai mari de 30 caractere etc.

Structura **BDR** (vezi fig. 2 cap. 3) se poate realiza direct in limbajul de descriere **LDD SOCRATE**, sau in limbajul de descriere corespunzător unui alt sistem de gestiune a datelor, preferat.

Mult mai interesantă ni se pare însă realizarea **BDR** cu ajutorul unui **limbaj intermediar** care să permită tratarea oricărei structuri și a oricăror proceduri de la un *nivel de inteligență superior* și prin care limbajul **INTELEC** să fie văzut ca un caz particular.

Acest limbaj intermediar ar avea rolul unui *metalimbaj* pentru **INTELEC**, putind realiza, ceea ce cerecătorii în inteligență artificială numesc *un al doilea nivel de decizie* ([66], p. 131 – 158 și [41]) capabil să evaluate performanțele primului nivel, și, pe baza unor criterii să modifice structura și procedurile acestuia din urmă. Fie $L(G)$ limbajul **INTELEC** (vezi cap. 3) și $\lambda(\gamma)$ un limbaj intermediar funcțional care lucrează cu entități proprii ϵ , are o gramatică proprie γ , poate da naștere la structuri proprii $\sigma(\epsilon)$ și în care există anumite proceduri proprii π de creare și modificare a structurilor, de actualizare a bazei de cunoștințe, de selecție a datelor după anumite criterii etc.

Structura $S(E, F)$ a entităților din limbajul **INTELEC** (vezi fig. 2, cap. 3) poate fi definită prin intermediul limbajului $\lambda(\gamma)$, utilizind entitățile proprii ale acestuia și o parte π , din procedurile π și anume procedurile destinate creării structurilor $\sigma(\epsilon)$. În acest fel, structura $S(E, F)$ a limbajului **INTELEC** va fi un caz particular de structură σ , să spunem $\sigma_1(\epsilon)$, creată în limbajul $\lambda(\gamma)$.

În mod analog, orice cunoștință particulară introdusă în cadrul structurii $S(E, F)$ ca entitate E sau frază F generată cu ajutorul gramaticii G (vezi cap. 3) va putea fi interpretată drept informație introdusă direct în structura $\sigma_1(\epsilon)$ și va putea fi manipulată ca atare cu ajutorul celeilalte părți π_m a procedurilor π , destinate manipulării entităților din baza de date.

Această dublă interpretare, creează posibilitatea construirii unei logici speciale de *evaluare automată a performanțelor* limbajului $L(G)$ (contabilizare a numărului de entități existente în bază și aparținind anumitor domenii care interesează conducerea operativă a producției, numărul de fraze legate de acestea, numărul de legături, frecvența folosirii lor etc.) transparentă pentru utilizator. În funcție de rezultatul evaluărilor, pot fi imaginate *proceduri automate de modificare a structurii* existente $\sigma_1(\epsilon)$ într-o structură $\sigma_2(\epsilon)$ mai apropiată de necesitățile practice de utilizare în anumite domenii a limbajului $L(G)$, apoi, în mod analog, se poate ajunge la $\sigma_3(\epsilon)$ și.m.d., cu condiția de a formula clar obiectivele urmărite.

Limbajul funcțional $\lambda(\gamma)$ este construit la rîndul său pe baza unui limbaj procedural complet, cu ajutorul căruia se pot realiza algoritmii menționati mai sus, conlucrarea între limbajul procedural și cel funcțional derivat fiind permanentă.

O asemenea soluție a fost experimentată de autori pe calculatorul **FELIX-C512** utilizind sistemul de gestiune a bazelor de date și limbajul **SOCRATE**. Cu ajutorul acestuia din urmă a fost construit un sistem de gestiune propriu, numit sistem de bază de date cu *structură generalizată* (SBDG) care, de fapt, este un limbaj funcțional de tip $\lambda(\gamma)$. Aceasta permite crearea, în mod conversațional, a oricărei structuri de date, modificarea

ei de la consola terminalului sau prin program, precum și optimizarea structurii — dacă se formulează criterii în acest scop. Se mai pot introduce și scoate date cu contabilizarea numărului de entități și caracteristici, prezente la un moment dat în bază, se pot executa tot felul de selecții ale entităților existente etc.

În sistemul SBDSC, structura particulară $\sigma_1(\varepsilon)$ menționată pentru modelul INTELEC poate fi realizată prin limbajul funcțional SBDSC și exploatață fie cu limbajul $L(G)$, fie independent de acesta. Este de reținut faptul că această soluție permite ca unele caracteristici esențiale ale bazei de cunoștințe, cum ar fi, de exemplu, coerenta informațiilor, completitudinea informațiilor în vederea rezolvării cazurilor cunoscute de conducere operativă a producției și.a., să poată fi verificate *independent* de folosirea limbajului $L(G)$, prin așa-zise *programe-demon* [73] construite în limbajul de bază și limbajul intermediar și utilizând structura generalizată. În acest fel baza de date relațională (BDR) poate fi ținută permanent la un nivel de coerentă satisfăcător, permitând eliberarea utilizatorilor de grija completării unor informații uzuale, curente, și facilitând concentrarea atenției asupra construirii și exploatarii procedurilor de conducere operativă a producției propriu-zise, așa cum s-a procedat de altfel și în lucrarea de față.

Dăm, pentru exemplificare, procedura de creare a structurii BDR pentru modelul INTELEC (vezi fig. 2, cap. 3) cu ajutorul SBDSC. Conversația are loc între utilizator (*u*) și calculator (*c*) la consola terminalului de teletransmisie. Comenzi necesare pentru manipularea SBDSC sunt rezumate în *Memento de utilizare* (vezi anexa).

Comenzi și răspunsuri la consola terminalului de teletransmisie

Note explicative

u : EXEC CRESTRP ?

Utilizatorul dorește să creeze o structură în baza de date. În acest scop apelează programul CRESTRP

c : ELEMENT DE STRUCTURA
(MAX. 20 CAR)

Se cere furnizarea numelui elementului de structură și se indică lungimea maxim admisă, în caractere

u : ENTITATE

Se indică numele primului element (fig. 2, cap. 3).

c : APARTINE DE ELEMENTUL

Se cere numele elementului de structură de nivel superior.

u :*

În cazul de față, fiind vorba de un element independent (situat pe primul nivel) se indică „*”.

c : TIP (0=ENTITATE/1=CAR.
NUM./2=CAR.CUVINT/3=REF/
4 = INEL)

Se cere tipul acestui element. Se menționează tipurile posibile (vezi MEMENTO de utilizare)

Se răspunde „entitate”

u : 0

Entitatea are cheie de regăsire? Răspunsurile posibile sunt indicate în paranteză.

c : CHEIE (0=NU/1=CH.NUM/
2 = CH. CUV.)

u : 1

c : LIM. INF. A CHEII ($>=$)

u : 1

c : LIM. SUP. A CHEII ($<=$)

u : 1000

c : ELEMENT DE STRUCTURA
(MAX. 20 CAR.)

u : DEN

c : APARTINE DE ELEMENTUL

u : ENTITATE

c : TIP (0=ENTITATE/1=CAR. NUM./
2=CAR. CUVINT/3=REF/4=INEL)

u : 2

c : CHEIE (0=NU/1=CH.NUM./
2 = CH. CUV)

u : 0

c : VECTOR (0=NU/1=DA)

u : 1

c : LUNGIMEA SIRULUI DE CARACTERE
(1-25, 30/0 = LUNG. VALABILA)

u : 0

c : CU INDEXARE (0=NU/1=DA)

u : 0

c : ELEMENT DE STRUCTURA
(MAX. 20 CAR.)

^{*)} Cheia este unică în sensul că nu pot exista în același timp în baza de date două rea-
lizări ENTITATE cu aceeași cheie.

Se răspunde „cheie numerică”.
— Vezi considerațiile din cadrul
cap. 3.

Se cere limita inferioară a cheii
numerice posibil a fi înregistrată
în baza de date pentru o ENTIT-
ATE oarecare *).

Se răspunde 1.

Se cere limita superioară a cheii
valabile pentru ENTITATE.

Se răspunde 1000. Se limitează
astfel *deocamdată*, numărul entită-
ților la această valoare. Ulterior
în caz de necesitate, prin conver-
sație, sau prin program, *această
limită poate fi mărită*.

Se trece la înregistrarea următo-
rului element de structură :
denumirea entității (DEN).

Se cere tipul.

Este vorba de o caracteristică tip
„cuvînt” (șir de caractere).

Se cere cheia.

Nu este nevoie de cheie. Accesul
se face prin entitate.

Denumirea este formată din mai
multe cuvînte. Dacă fiecare cu-
vînt este un element, ansamblul
formează un vector.

Se cere lungimea sirului de carac-
tere al fiecărui cuvînt. Se poate
indica o lungime fixă, începînd
cu 1 caracter pînă la 25 caractere
sau exact 30 caractere. Cifra 0 re-
prezintă cuvînte cu lungime vari-
abilă. În acest caz lungimea ur-
mează a fi precizată concomitent
cu inserierea cuvîntului în baza
de date.

Se trece la următorul element de
structură : caracteristica de in-
dexare (INDEX).

u : INDEX
 c : APARTINE DE ELEMENTUL
 u : ENTITATE
 c : TIP (0=ENTITATE/1 CAR. NUM./
 2=CAR. CUVINT/3=REF/4=INEL)
 u : 2

Este o caracteristică de tip „cuvint” formată din mai multe asemenea cuvinte „cheie” de lungime fixă, puse cap la cap și formând un vector.

c : CHEIE (0=NU/1=CH. NUM./
 2 = CH. CUV.)

Fiecare cuvint poate servi drept criteriu de regăsire a întregii entități ENTITATE.

Caracteristica, în ansamblu, nu este necesar să aibă cheie de regăsire deoarece este atașată entității ENTITATE dotată cu cheie, așa cum am văzut.
 Este de tip vector.

u : 0

Se alege lungimea maximă, de 30 caractere.

c : VECTOR (0=NU/1=DA)
 u : 1
 c : LUNGIMEA SIRULUI DE CARACTERE
 (1-25, 30/0=LUNG. VARIABILA)
 u : 30
 c : CU INDEXARE (0=NU/1=DA)
 u : 1

Fiecare sir de 30 caractere poate constitui un criteriu de regăsire a entității ENTITATE. Acest lucru se precizează chiar în momentul introducerii informației respective în baza de date.

c : ELEMENT DE STRUCTURA
 (MAX. 20 CAR.)
 u : NUMAR
 c : APARTINE DE ELEMENTUL
 u : ENTITATE
 c : TIP (0=ENTITATE/1=CAR.NUM./
 2=CAR. CUVINT/3=REF/4=INEL)
 u : 1
 c : CHEIE (0=NU/1=CH.NUM./
 2 = CH. CUV.)

Se creează caracteristica numerică NUMAR a entității ENTITATE (vezi fig. 2, cap. 3).

u : 0
 c : VECTOR (0=NU/1=DA)
 u : 0
 c : VALOARE LIM. INFER (> =)
 u : - 99999999

Caracteristica nu are cheie de regăsire (regăsirea se face prin ENTITATE).

Nu este de tip vector (reprezintă o valoare singulară).

Limita inferioară a caracteristicii este -99 999999.

c : LIM. SUPER (<=)

Limita superioară a caracteristicii este +99 999 999.

u :>9999999

e : ELEMENT DE STRUCTURA
(MAX. 20 CAR.)

Se creează entitatea LEG subordonată elementului ENTITATE (vezi fig. 2. cap. 3).

u : LEG

c : APARTINE DE ELEMENTUL

u : ENTITATE

c : TIP (0=ENTITATE/1=CAR.NUM./
2=CAR. CUV./3=REF/4=INEL)

u : 0

c : CHEIE (0=NU/1=CH. NUM./
2 = CH. CUV.)

u : 1

Cheia entității este numerică,
cuprinsă între 1 și 10 000.

c : LIM. INF. A CHEII (>=)

u : 1

c : 10000

e : ELEMENT DE STRUCTURA
(MAX. 20 CAR.)

Se creează caracteristica I de tip inel, subordonată entității LEG.

u : I

c : APARTINE DE ELEMENTUL

u : LEG

c : TIP (0=ENTITATE/1=CAR. NUM./
2=CAR. CUV./3 = REF/4=INEL)

u : 4

e : ELEMENT DE STRUCTURA
(MAX. 20 CAR.)

Se creează caracteristica R de tip referință, subordonată entității LEG.

u : R

c : APAETINE DE ELEMENTUL

u : LEG

c : TIP (0=ENTITATE (1=CAR. NUM./
2=CAR. CUV./3=REF/4=INEL)

u : 3

c : REFERA ELEMENTUL

u : I

ș.a.m.d.

Lista completă a acestei conversații tipărită pe consola CENTRONIX a unui terminal de teletransmisie legat la calculatorul FELIX—c512, este dată în ANEXA. Tot acolo este listată în continuare structura *BDR* realizată cu programul de mai sus și care reflectă concordanța rezultatului obținut față de schema din figura 2, capitolul 3.

Să introduceem în structura de mai sus, cu ajutorul aceluiași instrument informatic, o frază. Operația se realizează tot de la consola terminalui, răspunzând la întrebările puse de calculator.

u : EXEC CREATBAZAP ?

Se activează programul de introducere și actualizare a datelor în bază.

c : z1 :

Calculatorul așteaptă denumirea elementului de structură care corespunde informației ce va fi actualizată.

u : ENTITATE

Ne referim în continuare la elementul ENTITATE.

*c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)*

Calculatorul verifică existența elementului cu acest nume în structură și îl validează. Se poate trece la efectuarea operației (creare, actualizare, stergere) specificată de către utilizator.

Se dorește crearea.

*u : 1
c : CHEIA (1—999)
u : 217*

Se cere cheia entității care urmează a fi creată. În structură se vede că este vorba de o entitate cu cheie numerică. Tot acolo sunt inscrise și limitele între care trebuie să se găsească cheia furnizată de către utilizator.

Se comunică numărul 217 care este acceptat. Cheia este unică în sensul că nu pot exista simultan în bază două elemente de același fel cu aceeași cheie.

Se trece din nou comanda utilizatorului.

c : z1 :

Acesta răspunde că dorește să coboare un nivel în structura elementului ENTITATE creat anterior.

*u : —
c : SE COBOARA UN NIVEL IN
STRUCTURA*

Se indică elementul DEN (denumirea entității).

*c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE /2 = ACTUALIZARE/3=
STERGERE)*

Crearea denumirii.

*u : 1
c : CREARE DE LA INCEPUT (0)
SAU IN CONTINUARE (1)*

Fiind vorba de un vector, acesta poate fi creat începând cu primul său element, apoi al doilea, al treilea și a.m.d. (0) sau elementele care se creează se adaugă la cele

u : 0

c : CU INDEXARE (0=NU/1=DA)

u : 0

c : SE CONTINUA CU COMANDA +

z1 :

u : +

c : SIR DE CARACTERE

u : CONDUCEREA

c : LUNGIMEA

u : 10

c : z1 :

u : +

c : SIR DE CARACTERE

u : OPERATIVA

c : LUNGIMEA

u : 9

c : z1 :

u : +

c : SIR DE CARACTERE

u : A

c : LUNGIMEA

u : 1

c : z1 :

u : +

c : SIR DE CARACTERE

u : PRODUCTIEI

c : LUNGIMEA

u : 10

c : z1 :

u : INDEX

c : CODUL OPERATIEI (0=INEFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)

u : 1

c : CREARE DE LA INCEPUT (0) SAU
IN CONTINUARE (1)

u : 0

c : CU INDEXARE (0=NU/1=DA)

existente pînă în acel moment în bază (1).

Se indică pornirea de la zero.

Fiecare element creat poate constitui o cheie de regăsire (index). Dacă se acceptă această opțiune atunci cuvintele cheie urmează a fi specificate cu ocazia introducerii lor în bază.

Fără indexare.

Se introduce textul :

CONDUCEREA OPERATIVA A PRO-
DUCTIEI.

Se trece la următoarea caracteristică INDEX care constă din punerea cap la cap, sub formă de vector, a unor siruri de 30 caractere formînd cuvinte-cheie pentru ENTITATE.

Crearea indexului.



u : 1
 c : SE CONTINUA CU COMANDA +
 z1 :
 u : +
 c : SIR DE 30 CARACTERE
 u : CONDUCERE
 c : INDEXAT (0=NU/1=DA)
 u : 1
 c : z1 :
 c : +
 c : SIR DE 30 CARACTERE
 u : CONDUCEREA PROducTIEI
 c : INDEXAT (0=NU/1=DA)
 u : 1
 c : z1 :
 u : +
 c : SIR DE 30 CARACTERE
 u : CONDUCERE OPERATIVA
 c : INDEXAT (0 = NU/1 = DA)
 c : z1 :
 u : +
 c : SIR DE 30 CARACTERE
 u : INTREPRINDERE
 c : INDEXAT (0=NU/1=DA)
 c : z1 :
 u : TIP

 c : CODUL OPERATIEI (0=INEFFECTIV/
 1 = CREARE/2 = ACTUALIZARE/
 3 = STERGERE)

Se optează pentru indexare.

S-au introdus următoarele evante cheie : CONDUCERE, CONDUCEREA PROducTIEI, CONDUCERE OPERATIVA, INTREPRINDERE

În continuare se actualizează caracteristica TIP.

Entitatea este de tip „noțiune”

Se creează o subentitate de legătură LEG

u : 1
 c : SIR DE 1 CARACTERE
 u : N
 c : z1 :
 u : —

 c : SE COBOARA UN NIVEL IN
 STRUCTURA
 z1 :
 u : LEG
 c : CODUL OPERATIEI (0=INEFFECTIV/
 1 = CREARE/2 = ACTUALIZARE/3 =
 STERGERE)
 u : 1
 c : CHEIA (1—9999)
 u : 1911

 c : z1 :
 u : —
 c : SE COBOARA UN NIVEL IN STRUCTURA
 z1 :

Se creează entitatea de legătură 1911.

Se inscrie caracteristica DATA.

u : DATA
 c : VALOARE (800101—991231)
 u : 820301
 c : Z1 :
 u : IE

Se trece data creării textului legat de entitate:
 1 martie 1982.
 Se creează inelul de text.

c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

u : 1
 c : Z1 :
 u : AAA
 c : INEXISTENT

Această comandă este dată pentru a reveni la primul nivel în structură.

c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

u : 1
 c : CHEIA (1—9999)
 u : 111

Se dorește înscrierea unei fraze în inelul de text creat anterior, la capitolul CONDUCEREA OPERATIVA A PRODUCTIEI

Fraza va avea cheia de regăsire 111.

c : Z1 :
 u : —
 c : SE COBOARA UN NIVEL IN STRUCTURA
 u : IF
 c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

Se creează inelul frazei.
 De acesta vor fi legate toate entitățile care compun fraza:

u : 1
 c : Z1 :
 u : RE
 c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

u : 1
 c : REFERA IE AL ENTITATII LEG AVIND
 CHEIA

u : 1911

c : Z1 :
 u : AAA

c : INEXISTENT
 Z1 :

u : ENTITATE

c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

Prima noțiune care intră în compunerea frazei este <CONDUCEREA OPERATIVA A PRODUCTIEI>. Deoarece există deja în bază, se va crea numai entitatea de legătură.

u : 2
c : CHEIA (1-999)
u : 217
c : Z1 :
u : —
c : SE COBOARA UN NIVEL IN STRUCTURA
u : LEG
c : CODUL OPERATIEI (0=INEFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)

u : 1
c : CHEIA (1-9999)

Se creează entitatea de legătură 1912.

c : Z1 :
u : —
c : SECBOARA UN NIVEL IN STRUCTURA
Z1 :
u : RF
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)
u : 1
c : REFERA IF AL ENTITATII FRAZA
AVIND CHEIA

Se leagă la fraza 111.

c : Z1 :
u : I
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3 =
STERGERE)

Se trece la legarea celei de-a doua entități în cadrul frazei 111 și anume a predicatului.

Z1 :
u : ENTITATE
c : CODUL OPERATIEI (0=INEFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)

[ESTE CAZ PARTICULAR AL]

Presupunem că entitatea există deja în baza de date și are cheia 93. Se creează numai legătura corespunzătoare printr-o entitate LEG.

```

u : 2
c : CHEIA (1-999)
u : 93
c : Z1 :
u : -
c : SE COBOARA UN NIVEL IN
      STRUCTURA

```

u : LEG
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)
u : 1
c : CHEIA (1-9999)
u : 1913

Se creează entitatea de legătură 1913.

c : Z1 :
u : —
c : SE COBOARA UN NIVEL IN
STRUCTURA
u : RF
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)
u : 1
c : REFERA IF AL ENTITATII FRAZA
AVIND CHEIA

Se leagă la fraza 111.

u : 111
c : Z1 :
u : R
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)
u : 1
c : REFERA I AL ENTITATII LEG
AVIND CHEIA
u : 1912
c : Z1 :
u : I
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)
u : 1
c : Z1 :
u : AAA
c : INEXISTENT
Z1 :

Se creează legătura de precedență la entitatea anterioară în cadrul frazei.

u : ENTITATE
c : CODUL OPERATIEI (0=INEFFECTIV/
1=CREARE/2=ACTUALIZARE/3=
STERGERE)

Se trece la efectuarea ultimei legături și anume la entitatea **«PROCES DE CONDUCERE»** care presupunem că se află în bază și are cheia 175. Se creează legătura LEG corespunzătoare la fraza 111 și precedență cu entitatea anterioară.

u : 2
c : CHEIA (1-999)
u : 175
c : Z1 :
u : —
c : SE COBOARA UN NIVEL IN
STRUCTURA



u : LEG
 c : CODUL OPERATIEI (0=INEFFECTIV/
 · 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)
 u : 1
 c : CHEIA (1-9999)
 u : 1914
 c : Z1 :
 u : —
 c : SE COBOARA UN NIVEL IN
 STRUCTURA
 u : RF
 c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

u : 1
 c : REFERA IF AL ENTITATII FRAZA
 AVIND CHEIA

u : 111
 c : Z1 :
 u : R
 c : CODUL OPERATIEI (0=INEFFECTIV/
 1=CREARE/2=ACTUALIZARE/3=
 STERGERE)

u : 1
 c : REFERA I AL ENTITATII LEG
 AVIND CHEIA

Se leagă la fraza 111.

Se creează legătura de precedență
in cadrul frazei.

u : 1913
 c ; Z1 :
 u : *

În acest fel a fost inscrisă în BDR (vezi capitolul 3, paragraful 3.6. punctul b.4) **⟨CONDUCEREA OPERATIVA A PRODUCTIEI⟩** [ESTE (CAZ PARTICULAR AL) **⟨PROCES DE CONDUCERE⟩**].

Lista obținută pe calculator este dată în *anexă*.

Așa dar, ori de câte ori se vor utiliza entități existente în baza de date pentru a forma noi fraze, se vor crea subentități LEG corespunzătoare care vor fi atașate frazelor respective prin legături referință-inel RF-IF; Conexiunea „de ordine” (precedență) în cadrul frazei va fi realizată prin legături referință-inel de tipul R-I. De asemenea, pentru a putea stringe la un loc frazele referitoare la un domeniu exprimat printr-o entitate, se vor utiliza legăturile referință-inel de tipul RE-IE. În acest fel, baza de date relațională descrisă în cadrul capitolului 3 va fi utilizată în mod complet și rațional.

Așa cum am văzut, nu orice înșiruire de entități reprezintă o frază. Regulile corespunzătoare formării unei fraze în limbaj INTELEC au fost enunțate cu ocazia definirii gramaticii G a acestui limbaj (cap. 3, prg. 3.2.) Programul care, plecind de la un sir de entități date, stabilește dacă acest sir aparține sau nu limbajului $L(G)$ poartă numele de *analizor sintactic*. O frază oarecare, aparținând limbajului $L(G)$ este executabilă, dacă con-

ține un predicat intern executabil (de tipul Q vezi cap. 3). În acest caz, execuția constituie o fază separată, în cadrul căreia are loc identificarea șirului de comenzi interne corespunzător înțelesului frazei (corespondență semantică) și îndeplinirea acestor comenzi. Cu alte cuvinte, dacă predi-



Fig. 27. – Succesiunea fazelor de utilizare a modelului INTELEC.

catul nu este intern, fraza rămîne înscrisă în baza de date și constituie o indicație, un control, o înregistrare a unui fapt sau pur și simplu o informație de un anumit nivel pentru utilizator. Dacă însă predicatul este intern, fraza reprezintă un *program* care poate fi executat de către calculator.

Fraza executabilă este transformată în program executabil de către un *analizor semantic* care identifică șirul de comenzi interne corespunzător predicatului intern și le aplică asupra subiectului sau complementului intern (de tipul K) – vezi schema din figura 27.

Cele două faze – înregistrarea și execuția – sunt independente dar ordinea lor este obligatorie. Analiza sintactică și recunoașterea textului formulat de utilizator ca aparținind limbajului $L(G)$ au loc cu ocazia tentativei de înscriere a acestui text în BDR. Conform schemei logice din figura 28, dacă fraza este corectă se admite înscrierea. Dacă nu este corectă, se cere reformularea sau completarea ei cu informațiile care lipsesc. Execuția aceluiași text poate avea loc imediat după înregistrare sau oricând după acest eveniment. Așa cum se indică în schema logică din figura 29, este necesară mai întâi *delimitarea* textului (ca aparținind unei anumite entități sau prin indicarea numerelor de secvență ale frazelor – vezi cap. 3). În urma analizei semantice se ajunge fie la selecția și execuția programelor interne, fie numai la simpla listare a frazelor la terminal.

Pentru limbajul INTELEC, analizorul semantic poate fi realizat sub forma unor *tabele de decizie* (vezi cap. 2) cu mai multe intrări, rezultatul – programul execu-

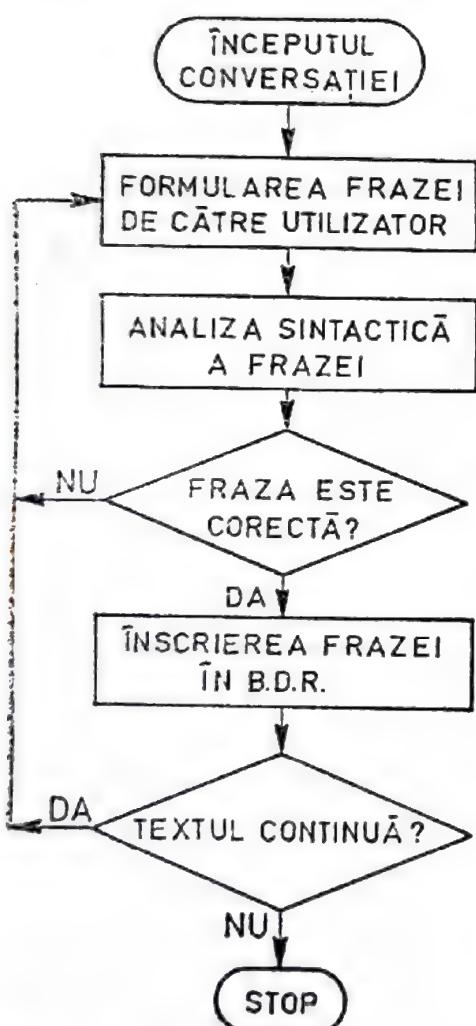


Fig. 28. – Schema logică de înregistrare în BDR a unui text formulat în limbaj L (faza I).

bil – fiind unic. Se poate folosi în acest scop însuși limbajul de manipulare al bazei de date (de exemplu LMD SOCRATE) sau module speciale scrise în limbajul de asamblare (de exemplu ASSIRIS).

În ceea ce privește analizorul sintactic, acesta poate fi realizat, la rîndul său în limbajul de manipulare al bazei de date, cu module în limbaj de asamblare sau, în orice alt limbaj ca fază independentă și premergătoare analizei semantice.

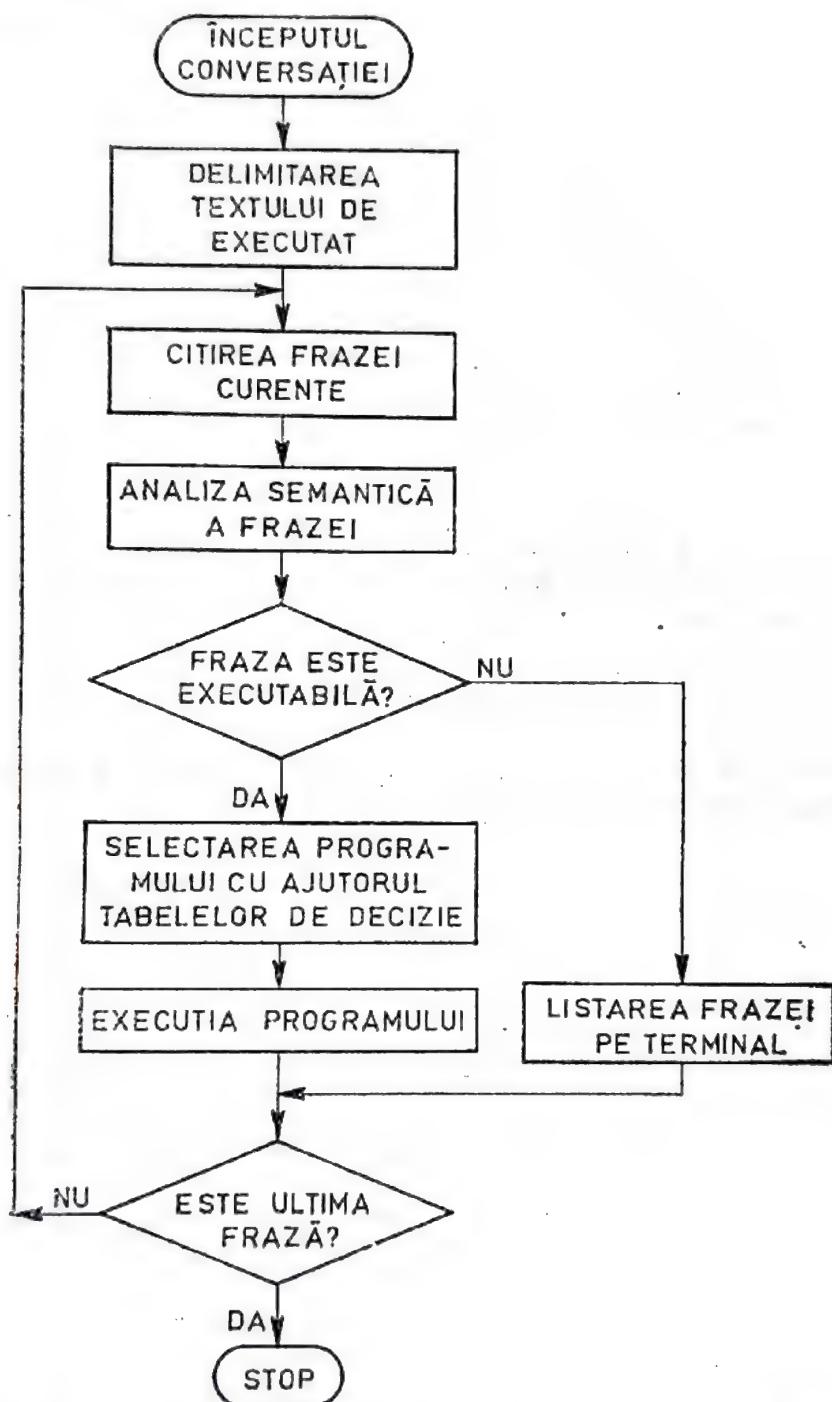


Fig. 29. — Schema logică de parcurgere și executare a unui text înscris în BDR (faza a II-a).

O soluție interesantă o constituie realizarea analizorului sintactic cu ajutorul limbajelor de tip LISP ([73] p. 257—375) sau a celor derivate (MACLISP, INTERLISP) [61, 71]. Caracteristica principală a acestor limbaje

este posibilitatea de *recursivitate* a definițiilor. Limbajul LISP conține funcții de bază deosebit de eficace în manipularea sirurilor de caractere (CAR, CDR, APPEND și CONS) precum și funcții speciale (MATCH) care permit recunoașterea unor sevențe de cuvinte (entități).

O primă fază a analizei sintactice ar putea face, în acest caz, trecrea de la scrierea într-un limbaj apropiat de cel natural, cu multe articole, numerale etc., la un limbaj relațional, ca de exemplu INTELEC, limitând frazele la nucleul lor prin izolarea cuvintelor „cheie”.

Utilizarea funcției MATCH împreună cu semnele speciale „?” și „*” ar ușura rezolvarea problemei. Menționăm că semnul „?” intercalat în prototip înlocuiește orice cuvânt din sirul inseris de utilizator. De exemplu, comparațiile următoare sunt adevărate, programul returnind valoarea *T* (adevărat) pentru orice combinație de două cuvinte în care primul are valoarea 'PRODUCȚIA'.

```
(MATCH '(PRODUCTIA ?) '(PRODUCTIA INDUSTRIALA)
      T
      (MATCH '(PRODUCTIA ?) '(PRODUCTIA INTREPRINDERII)
      T
      etc.
```

Semnul „*” are rolul de a extinde valabilitatea comparației la un întreg sir de cuvinte care preced sau urmează cuvântului „cheie” menționat. De exemplu :

```
(MATCH '(*PRODUCTIE*) '(OPERATIILE DE PRODUCTIE SE PROGRAMEAZA
      PE CALCULATOR)
      T
      (MATCH '(*PRODUCTIE*) '(REZULTATELE OBTINUTE IN PRODUCTIE SE
      COMPARA CU CELE CALCULATE TEORETIC)
      T
      etc.
```

Toate frazele care conțin cuvântul „cheie” 'PRODUCTIE' sunt acceptate.

Prin utilizarea alternată a funcțiilor CAR (care detașează primul cuvânt din cele furnizate) și CDR (care ia în considerație restul de cuvinte, după ce primul cuvânt din listă a fost eliminat) se pot construi programe de analiză puternice. Se pot introduce, de asemenea, funcții speciale de restricție care permit ca în locurile marcate din prototip să fie acceptate cuvinte având anumite proprietăți. Forma acestor cuvinte este definită prin funcția RESTRICT, astfel :

```
(RESTRICT ?
      <proprietate 1>
      <proprietate 2>
      . . . . .)
```

Utilizarea acestor funcții mărește puterea de analiză a programului. Fie, de exemplu :

```
(MATCH '(*RESTRICT ? PREDICAT*)
      '((CONDUCEREA PRODUCTIEI) (SE REFERA LA) (INTREPRINDERE))
      T
```

în care 'PREDICAT' este numele unei funcții care verifică faptul că proprietatea TIP a entității analizate este 'P', cu alte cuvinte, că este vorba de

un predicat (vezi subcap. 3.4). Se vede că programul de mai sus admite toate frazele care au un predicat. Tot aşa pot fi alcătuite programe care pun în evidență frazele construite cu anumite predicate (de exemplu : [SE COMPUNE DIN], [ESTE CAZ PARTICULAR AL] etc.) făcind astfel posibilă „înțelegerea” și clasificarea lor.

Elementele de structură ale frazei în limbaj INTELEC odată separate, se poate trece, în *faza a doua*, la verificarea sintaxei limbajului.

Utilizând facilitățile puse la dispoziție de LISP, verificarea grupului de axiome I – VI ale limbajului INTELEC este simplă. Să vedem, de exemplu, cum am realizat în LISP verificarea producțiilor *e*, *f* și *g* care definesc formal grupul *<Notiune>* (vezi subcap. 3.2.).

Notății : *S* = lista de cuvinte de analizat, *W* = cuvântul curent

(**DEFINE (NOTIUNE)**)

```

(PROG (GRUP EXPRESIE SIR)
      (SETQ EXPRESIE S W(CAR S) GRUP (GENNAME 'NOTIUNE))
      (SETQ IND1 NIL IND2 NIL)
      PUTF GRUP DEN
      PUTF GRUP TIP
      C1 (COND ((MEMBER 'N (GET W 'TIP))
                 (SETQ W (CAR (SETQ S (CDR S))))
                 (CONS W SIR)
                 (GO C12))
                 ((T GO ESEC)))
      C12 (COND ((MEMBER 'C (GET W 'TIP))
                  (SETQ W (CAR (SETQ S (CDR S))) IND1 T)
                  (CONS W SIR)
                  (GO C12))
                  ((AND (IND1)
                        (MEMBER 'N (GET W 'TIP)))
                   (SETQ W (CAR (SETQ S (CDR S)))))
                   (CONS W SIR)
                   (SETQ IND1 NIL IND2 T)
                   (GO C12))
                  ((OR (IND2)
                       (AND (NOT (IND1)) (IND2))))
                   (GO REUSITA))
                  ((T GO ESEC)))
      REUSITA (PUTPROP GRUP SIR 'DEN)
              (PUTPROP GRUP TIP 'N)
              (SETQ ULTIM GRUP)
      ESEC (SETQ S EXPRESIE W NIL)
            (RETURN NIL)))

```

Programul de mai sus încearcă să izoleze un grup de entități din sirul *s*, grup care să aibă proprietățile exprimate de relațiile *e*, *f* și *g* (cap. 3, prg. 3.2). În caz de reușită, grupul de entități este returnat în *SIR* și va purta (temporar) numele *NOTIUNEN* care are la sfîrșit un număr de ordine *n* pentru cazul în care fraza ar conține eventual mai multe grupuri de același tip (proprietatea funcției speciale *GENNAME*). În caz de eșec, sirul *s*, memorat la începutul programului în *EXPRESIE*, este returnat cu valoarea inițială, împreună cu evaluarea *NIL* (fals) pentru întreaga funcție

NOTIUNE : Funcția PUTF adaugă caracteristicile DEN, TIP etc. entității noi create GRUP. În paragraful 'c1' se verifică prin COND (condiție) că prima entitate izolată în w este de TIP 'N' (vezi regula e).

Dacă acest lucru este adevărat se trece la paragraful 'c12'. În paragraful 'c12' se verifică următoarele entități care trebuie să fie, în cazul în care există, un *<Grup CN>* (vezi cap. 3) format dintr-un calificator ('c') și o notiune ('N'). Utilizarea adesea a indicatorilor IND1 și IND2 permite selectarea acestei posibilități.

În mod analog se construiesc și celelalte funcții pentru verificarea grupurilor *<Predicat>*, *<Grup ON>*, *<Comandă>* și *<Comandă condiționată>* care intră în alcătuirea limbajului (vezi cap. 3).

La nivelul superior, asamblarea acestor funcții are loc conform producției c (vezi subcap. 3.2.). O realizare posibilă a acesteia prin LISP este dată mai jos :

```
(DEFINE (FRAZA))
  (PROG (COMPILATOR INTELEC)
    A (COND ((NOTIUNE)
              (PREDICAT)
              (GO A1))
              ((NOTIUNE)
               (GRUP ON)
               (GO A1))
              ((AND (COMANDA) (NULL S)) (RETURN T))
              ((AND (COMANDA CONDITIONATA)
                    (NULL S))
               (RETURN T))
              ((T (RETURN NIL))))
    A1 (COND ((PREDICAT)
              (NOTIUNE)
              (GO A2))
              (GRUP ON)
              (PREDICAT)
              (GO A2))
              ((T (RETURN NIL))))
    A2 (COND ((AND (NOTIUNE) (NULL S)) (RETURN T))
              ((NOTIUNE)
               (GRUP ON)
               (GO A3))
              ((PREDICAT)
               (NOTIUNE)
               (GO A3))
              ((T (RETURN NIL))))
    A3 (COND ((AND (GRUP ON) (NULL S)) (RETURN T))
              ((AND (NOTIUNE) (NULL S)) (RETURN T))
              ((T (RETURN NIL)))))
```

De fiecare dată cind se reușește identificarea unui grup a cărui denumire este simbolizată prin numele funcției, valoarea returnată este T (adevărat). Această valoare reprezintă condiția (exprimată prin COND), de executare a funcției următoare. De exemplu, la paragraful A dacă (NOTIUNE) returnează T atunci se execută (PREDICAT) după care se trece la paragraful A1. La paragraful A1, dacă (PREDICAT) a returnat T atunci se

execută din nou (NOTIUNE) și se trece mai departe la paragraful A2. Aici se verifică dacă identificarea grupului *<Noțiune>* a reușit. În acest caz se verifică terminarea șirului *S*. Dacă funcția (NULL S) este și ea adevărată analiza se termină cu bine. Fraza pusă în evidență este de tipul

<Noțiune> <Predicat> <Noțiune>

Dacă șirul *S* nu s-a terminat atunci se încearcă mai departe identificarea unui *<Grup ON>* ajungind la paragraful A3. Dacă ultimul grup a fost intr-adevăr *<GRUP ON>* și s nu mai are nici o entitate atunci analiza reușește și pune în evidență o frază de tipul :

<Noțiune> <Predicat> <Noțiune> <Grup ON>

s.a.m.d.

Mentionăm, în încheiere, că implementarea funcțiilor de mai sus trebuie să facă apel la proceduri rapide de regăsire, în conformitate cu sistemul de bază de date utilizat pentru depozitarea informațiilor. Aceste proceduri sint, de regulă, realizate prin programe scrise în limbajul de bază al calculatorului.

A N E X A

Această anexă cuprinde :

1. Memento de utilizare a produsului informatic denumit **SISTEM DE BAZA DE DATE CU STRUCTURA GENERALIZATA (S.B.D.S.G.)**, operațional pe calculatorul **FELIX-C512**.

2. Crearea, cu ajutorul S.B.D.S.G., structurii Bazei de Date Relaționale (BDR) precum și introducerea, pe această structură, a entităților :

⟨COND. OPERATIVA A PRODUCTIEI⟩ — tip *noțiune*, indexată pe următoarele cuvinte cheie de regăsire : **CONDUCERE, CONDUCEREA PRODUCTIEI, COND. OPERATIVA A PRODUCTIEI, INTREPRINDERE** ;

[ESTE CAZ PARTICULAR AL] — tip *predicat*, indexat pe cuvîntul cheie **PREDICAT** ;

⟨PROCES DE CONDUCERE⟩ — tip *noțiune*, indexat pe cuvintele cheie de regăsire **PROCES ȘI CONDUCERE**.

3. Înscrierea textului cu titlul **COND. OPERATIVA A PRODUCTIEI** și, în cadrul acestui text, înregistrarea următoarei fraze în limbaj INTELEC :

⟨COND. OPERATIVA A PRODUCTIEI⟩ [ESTE CAZ PARTICULAR AL] ⟨PROCES DE CONDUCERE⟩.

Rezultatele sunt obținute în regim de teleprelucrare, cu sistemul **SOCRATE-ITC, VER. 04/05/83**, pe un terminal de tip **CENTRONICS** conectat la Centrul de calcul al Academiei „**Ștefan Gheorghiu**”.

Atragem atenția cititorului că, ANEXA fiind de dată mai recentă, între textul principal al cărții și rezultatele prezentate există unele diferențe. Pentru a veni în ajutorul cititorului și a facilita înțelegerea conversației cu calculatorul, în partea dreaptă a paginii sunt inscrise cifre care corespund unor explicații sumare date la sfîrșitul listei.

MEMENTO DE UTILIZARE AL S.B.D.S.G. SISTEM DE BAZĂ DE DATE CU STRUCTURA GENERALIZATĂ PENTRU MODELUL INTELEC

Exemplu

1. STRUCTURA	Răspuns	Semnificație
ELEMENT DE STRUCTURA (max. 20 car.)	FRAZA	—
APARTINE DE (denumirea elementului de care aparține)	*	primul nivel
TIP 0 — entitate		
1 — caracteristică numerică		
2 — caracteristică sir de caracter		
3 — referință		
4 — inel	0	entitate
Pentru TIP = 0,1,2		
CHEIE 0 — fără cheie		
1 — cheie numerică		
2 — cheie sir de caractere	1	cheie numerică

Pentru CHEIE = 1
 LIM. INF. A CHEII (\geq) 1
 LIM. SUP. A CHEII (\leq) 50000
Pentru CHEIE = 2
 LUNGIMEA CHEII (max. 30 car.)
Pentru CHEIE = 0
 VECTOR | 0 — nu
 | 1 — da
Pentru TIP = 1
 VAL. LIM. INF. (intre —99999999 și 1
 99999999)
 VAL. LIM. SUP. (idem)
 NR. ZECIMALE (max. 9)
Pentru TIP = 2
 LUNG. SIR DE CARACTERE (1 — 25,30/
 0 = LUNG. VARIABILA)
Pentru TIP = 3
 REFERA ELEMENTUL (denumirea elemen-
 tului pe care îl referă)

Exemplu

2. INTRODUCEREA DATELOR :	Răspuns	Semnificație
Denumirea elementului de structură		FRAZA
CODUL OPERATIEI 0 — inefectiv 1 — creare 2 — actualizare 3 — ștergere	1	creare
<i>Pentru TIP = 0,1,2 și CHEIE = 1</i> CHEIA (lim.inf.—lim.sup.)	3	FRAZA cu cheia = 3
<i>Pentru TIP = 0,1,2, CHEIE = 2</i> CHEIA (lungime)		
<i>Pentru TIP = 0,1,2, CHEIE = 0</i> VECT = 1 și CODUL OPERATIEI = 2		
NUMARUL ELEM. IN CADRUL VECTORULUI		
<i>Pentru TIP = 0,1,2 CHEIE = 0</i> VECT = 1 și CODUL OPERATIEI = 1		
CREARE DE LA ÎNCEPUT (0) SAU		
IN CONTINUARE LA CEEA CE EXISTA (1)		
CU INDEXARE (1) SAU FARA (0)		
<i>Pentru TIP = 1</i> VALOARE (lim. inf.-lim. sup.)		
<i>Pentru TIP = 2</i> SIR DE CARACTERE (lungimea)		
<i>Pentru TIP = 3</i> REFERA ELEMENTUL CU CHEIA (lim.inf. — lim.sup.)		

Observații : 1. Caracteristicile de tip vector se leagă prin comanda +.
 2. Pentru caracteristicile cu lungime variabilă se furnizează în plus lungimea.

3. Pentru a ajunge la elementul situat pe nivelul n în structură se pornește de la entitatea situată pe primul nivel și se coboară de $n-1$ ori. Coborîrea unui nivel se realizează cu comanda —.

Exemplu

3. SELECTAREA INFORMATIILOR : Răspuns Semnificație

ENTITATE		FRAZĂ
CARACTERISTICA		
a. După caracteristică numerică		PF
LIM. INF. (\geq)		80
LIM. SUP. ($<$)		100
b. După caracteristică sir de caractere		CF
SIR DE CARACTERE		D
c. După caracteristică tip referință		RE
REFERA ENTITATEA CU CHEIA		917
d. După index		INDEX
SIR DE CARACTERE		CONDUCEREA OPERATIVA
CRITERIU INITIAL = 0		0 criteriu initial

CRITERIU CONSECUTIV LEGAT PRIN*SAU*=1

CRITERIU CONSECUTIV LEGAT PRIN*SI*=2

Observații: După fiecare selectare apare numărul entităților selectate pînă la momentul respectiv.

4. LISTAREA INFORMATIILOR SELECTATE
(FORMAT STANDARD) :

ENTITATE
COLOANA DE INCEPUT 1 (alinierea textelor cu format variabil și a denumirii caracteristicilor)
COLOANA DE INCEPUT 2 (alinierea conținutului caracteristicilor)

Observații: 1. Ordinea de listare a caracteristicilor în cadrul aceleasi entități este următoarea :

Din coloana 1 denumirea entității urmată de cod.

În continuare, din COLOANA DE INCEPUT 1 textul cu format variabil iar din COLOANA DE INCEPUT 2 caracteristicile sir de caractere cu lungime fixă, caracteristicile numerice și caracteristicile tip referință.

2. Există posibilitatea listării în ordinea crescătoare sau descrescătoare a codului (dacă acesta este numeric) și în ordinea crescătoare sau descrescătoare a unei caracte-ristici numerice.

CO
 QUESTION:CRESTR ?
 CREAREA UNUI ELEMENT DE STRUCTURA
 DENUMIRE (MAX.30 CAR.)MEMA
 SUBORDONAT ELEMENTULUIX
 TIP (0=ENT./1=CAR.NUM.INTR./2=CAR.NUM.ZEC./3=CAR.CUV./4=CAR.
 I+Z/5=CAR.I+CUV./6=CAR.Z+CUV./7=CAR.I+Z+CUV.,)
 Y2 :1
 LIMITA INFERIOARA-999999999
 LIMITA SUPERIOARA999999999
 OPTIUNI (0=FARA/1=REF.SIMPLA/2=REF.PE INEL/ 3 =INEL /4=REF.S
 +REF.PE I/5=REF.S+INEL/6=REP.PE I+INEL/7=REF.S+REF.PE I+INEL)
 Y7 :0
 CHEIE DE REGASIRE (0=FARA/1=CH.NUM./2=CH.CUVINT/3=CH.NUM.+CH.CUVINT)
 Y8 :1
 LIM.INF.A CHEII0
 LIM.SUP.A CHEIII100
 CHEIE UNICA (0=NU/1=CH.NUM.UNICA/2=CH.CUV.UNICA/3=AMBELE CHEI UNICE)
 Y12:1
 VECTOR (0=NU;1=DA)0
 CREAREA UNUI ELEMENT DE STRUCTURA
 DENUMIRE (MAX.30 CAR.)ENTITATE
 SUBORDONAT ELEMENTULUIX
 TIP (0=ENT./1=CAR.NUM.INTR./2=CAR.NUM.ZEC./3=CAR.CUV./4=CAR.
 I+Z/5=CAR.I+CUV./6=CAR.Z+CUV./7=CAR.I+Z+CUV.,)
 Y2 :5
 LIMITA INFERIOARA0
 LIMITA SUPERIOARA4
 LUNGIME CUVINT (1-30;0=VAR)1
 OPTIUNI (0=FARA/1=REF.SIMPLA/2=REF.PE INEL/ 3 =INEL /4=REF.S
 +REF.PE I/5=REF.S+INEL/6=REP.PE I+INEL/7=REF.S+REF.PE I+INEL)
 Y7 :0
 CHEIE DE REGASIRE (0=FARA/1=CH.NUM./2=CH.CUVINT/3=CH.NUM.+CH.CUVINT)
 Y8 :2
 LUNGIMEA.CHEII (1-30)30
 CHEIE UNICA (0=NU/1=CH.NUM.UNICA/2=CH.CUV.UNICA/3=AMBELE CHEI UNICE)
 Y12:2
 VECTOR (0=NU;1=DA)0
 CREAREA UNUI ELEMENT DE STRUCTURA
 DENUMIRE (MAX.30 CAR.)INDEX
 SUBORDONAT ELEMENTULUIENTITATE
 TIP (0=ENT./1=CAR.NUM.INTR./2=CAR.NUM.ZEC./3=CAR.CUV./4=CAR.
 I+Z/5=CAR.I+CUV./6=CAR.Z+CUV./7=CAR.I+Z+CUV.,)
 Y2 :3
 LUNGIME CUVINT (1-30;0=VAR)30
 OPTIUNI (0=FARA/1=REF.SIMPLA/2=REF.PE INEL/ 3 =INEL /4=REF.S
 +REF.PE I/5=REF.S+INEL/6=REP.PE I+INEL/7=REF.S+REF.PE I+INEL)
 Y7 :0
 CHEIE DE REGASIRE (0=FARA/1=CH.NUM./2=CH.CUVINT/3=CH.NUM.+CH.CUVINT)
 Y8 :0
 CHEIE UNICA (0=NU/1=CH.NUM.UNICA/2=CH.CUV.UNICA/3=AMBELE CHEI UNICE)
 Y12:0
 VECTOR (0=NU;1=DA)1
 CREAREA UNUI ELEMENT DE STRUCTURA
 DENUMIRE (MAX.30 CAR.)FRAZA
 SUBORDONAT ELEMENTULUIX
 TIP (0=ENT./1=CAR.NUM.INTR./2=CAR.NUM.ZEC./3=CAR.CUV./4=CAR.
 I+Z/5=CAR.I+CUV./6=CAR.Z+CUV./7=CAR.I+Z+CUV.,)
 Y2 :6
 NR.DE ZECIMALE (0-7)2
 LUNGIME CUVINT (1-30;0=VAR)1
 OPTIUNI (0=FARA/1=REF.SIMPLA/2=REF.PE INEL/ 3 =INEL /4=REF.S
 +REF.PE I/5=REF.S+INEL/6=REP.PE I+INEL/7=REF.S+REF.PE I+INEL)
 Y7 :0
 CHEIE DE REGASIRE (0=FARA/1=CH.NUM./2=CH.CUVINT/3=CH.NUM.+CH.CUVINT)
 Y8 :



Y8 :1
 LIM.INF,A CHEIII
 LIM.SUP,A CHEIII100
 CHEIE UNICA (0=NU/1=CH.NUM.UNICA/2=CH.CUV.UNICA/3=AMBELE CHEI UNICE)
 Y12:1
 VECTOR (0=NU;1=DA)0
 CREAREA UNUI ELEMENT DE STRUCTURA
 DENUMIRE (MAX.30 CAR.)LEG
 SUBORDONAT ELEMENTULUIENTITATE
 TIP (0=ENT./1=CAR.NUM.INTR./2=CAR.NUM.ZEC./3=CAR.CUV./4=CAR.
 I+Z/5=CAR.I+CUV./6=CAR.Z+CUV./7=CAR.I+Z+CUV.)
 Y2 :1
 LIMITA INFERIORA
 LIMITA SUPERIOARA999
 OPTIUNI (0=FARA/1=REF.SIMPLA/2=REF.PE INEL/3=INEL/4=REF.S
 +REF.PE I/5=REF.S+INEL/6=REF.PE I+INEL/7=REF.S+REF.PE I+INEL)
 Y7 :7
 REFERA SIMPLU ENTITATEAENTITATE
 REFERA PE INEL ENTITATEAFRAZA
 CHEIE DE NEGASIRE (0=FARA/1=CH.NUM./2=CH.CUVINT/3=CH.NUM.+CH.CUVINT)
 Y8 :1
 LIM.INF,A CHEIII800101000
 LIM.SUP,A CHEIII991231999
 CHEIE UNICA (0=NU/1=CH.NUM.UNICA/2=CH.CUV.UNICA/3=AMBELE CHEI UNICE)
 Y12:0
 VECTOR (0=NU;1=DA)1
 CREAREA UNUI ELEMENT DE STRUCTURA
 DENUMIRE (MAX.30 CAR.)
 \$GO
 QUESTION:POUR UN DICO AVEC DEN = 'FRAZA';
 * M REF = 'LEG'
 * M OPT = 6
 * FIN ?
 \$GO
 QUESTION:LISSTR ?
 ELEMENTUL (\$=TOATE)\$
 NIVEL1
 NR.DE ELEM.IGNOREATE

5

6

7

0 LISTAREA STRUCTURII

MAT1	CAR.NUM.INTR. -999999999-999999999
MEMA	FARA OPTIUNI CHEIE NUM. 0-100 UNICA
ENTITATE	CAR.NUM.INTR. -999509999-999999999
INDEX	FARA OPTIUNI CHEIE NUM. 0-100 UNICA
LEG	CAR.NUM.INTR. 0-999 VECTOR REF.SIMPLU/
FRAZA	ENTITATE REF.PE INEL/FRAZA ● NIEE NUM. 800101000-991231999
ELEMENTUL (\$=TOATE)\$	CAR.NUM.ZEC./2 ZECIMALECAR.CUVINT LG.CUV.= 1 REF.PE INEL/LEG CHEIE NUM. 1-100 UNICA
\$	

GO
 QUESTION:CREBAZA ?
 DENUMIRE IN STRUCTURAENTITATE
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CUVINT 30 CARACTERE
 Z2 :COND.OPERATIVA A PRODUCIEI
 CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
 Y10:1
 VALOARE INTREAGA (0=NU;1=DA)?
 Y10:1
 VALOARE 0-4
 Y4 :0
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 1 CARACTERE
 Z3 :N
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURA-
 SE COBOARA UN NIVEL IN STRUCTURA
 DENUMIRE IN STRUCTURAINDEX
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CREARE VECTOR (0=DE LA INDEPUT;1=IN CONTINUARE;2=INSERARE)?
 Y3 :0
 CU INDEXARE (0=NU;1=DA)?
 Y2 :1
 INITIALIZARE EFECTUATA, CONTINUATI CU COMANDA +
 DENUMIRE IN STRUCTURA+
 * CREARE (0=NU;1=DA) ?
 Y10:1
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 30 CARACTERE
 Z3 :CONDUCERE
 INDEXAT (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURA+
 * CREARE (0=NU;1=DA) ?
 Y10:1
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 30 CARACTERE
 Z3 :CONDUCEREA PRODUCIEI
 INDEXAT (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1

8

9

Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
Y10:1
* INREGISTRAT
DENUMIRE IN STRUCTURA+
* CREARE (0=NU;1=DA) ?
Y10:1
SIR DE CARACTERE (0=NU;1=DA)?
Y10:1
CUVINT 30 CARACTERE
Z3 :COND.OPERATIVA A PRODUCTIEI
INDEXAT (0=NU;1=DA)?
Y10:1
* CREAT
DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
Y10:1
* INREGISTRAT
DENUMIRE IN STRUCTURA+
* CREARE (0=NU;1=DA) ?
Y10:1
SIR DE CARACTERE (0=NU;1=DA)?
Y10:1
CUVINT 30 CARACTERE
Z3 :INTREPRINDERE
INDEXAT (0=NU;1=DA)?
Y10:1
* CREAT
DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
Y10:1
* INREGISTRAT
DENUMIRE IN STRUCTURAENTITATE
* CREARE (0=NU;1=DA) ?
Y10:1
CHEIA 30 CARACTERE
Z2 :ESTE CAZ PARTICULAR AL
CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
Y10:1
VALOARE INTREAGA (0=NU;1=DA)?
Y10:1
VALOARE 0-4
Y4 :0
SIR DE CARACTERE (0=NU;1=DA)?
Y10:1
CUVINT 1 CARACTERE
Z3 :P
* CREAT
DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
Y10:1
* INREGISTRAT
DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
Y10:

10

1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURA-
 SE COBOARA UN NIVEL IN STRUCTURA
 DENUMIRE IN STRUCTURAINDEX
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CREARE VECTOR (0=DE LA INCEPUT;1=IN CONTINUARE;2=INSERARE)?
 Y3 :0
 CU INDEXARE (0=NU;1=DA)?
 Y2 :1
 INITIALIZARE EFECTUATA, CONTINUATI CU COMANDA +
 DENUMIRE IN STRUCTURA+
 * CREARE (0=NU;1=DA) ?
 Y10:1
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:PREDICAT
 PR.MAIN *****ERREUR VALEUR NUMERIQUE ERROREE
 Y19:1
 CUVINT 30 CARACTERE
 Z3 :PREDICAT
 INDEXAT (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURAENTITATE
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CHEIA 30 CARACTERE
 Z2 :PROCES DE CONDUCERE
 CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
 Y10:1
 VALOARE INTREAGA (0=NU;1=DA)?
 Y10:1
 VALOARE 0-4
 Y4 :0
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 1 CARACTERE
 Z3 :N
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURAEN
 DENUMIRE IN STRUCTURA-
 SE COBOARA UN NIVEL IN STRUCTURA
 DENUMIRE IN STRUCTURAINDEX
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CREARE VECTOR (0=DE LA INCEPUT;1=IN CONTINUARE;2=INSERARE)?
 Y3 :

11

12

13



0
 CU INDEXARE (0=NU;1=DA)?
 Y2 :1
 INITIALIZARE EFECTUATA, CONTINUATI CU COMANDA +
 Denumire in structura+
 * CREARE (0=NU;1=DA) ?
 Y10:1
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 30 CARACTERE
 Z3 :PROCES
 INDEXAT (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 Denumire in structura+
 * CREARE (0=NU;1=DA) ?
 Y10:1
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 30 CARACTERE
 Z3 :CONDUCERE
 INDEXAT (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 Denumire in structuraentitate
 * CREARE (0=NU;1=DA) ?
 Y10:0
 * POZITIONARE/STERGERE
 SE INVESTIGHEAZA SUBANSAMBLUL 1
 CONTINE 10 ENTITATI
 ENTITATE CHEIA:COND.OPERATIVA A PRODUCTIEI VALDARE:NO
 CONTINUATI (0=NU;1=DA) ?
 Y10:1
 SE STERGE IN INTREGIME (0=NU;1=DA)
 Y1 :0
 SE POZITIONEAZA (0=NU;1=DA)
 Y1 :1
 ENTITATE CHEIA:ESTE CAZ PARTICULAR AL VALDARE:PO
 CONTINUATI (0=NU;1=DA) ?
 Y10:0
 Denumire in structura-
 SE COBOARA UN NIVEL IN STRUCTURA
 Denumire in structuraleg
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CHEIA800101000-991231999
 Y2 :830701000
 CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
 Y10:1
 VALOARE INTREAGA (0=NU;1=DA)?
 Y10:

14

195

1°
 VALDARE 0-999
 Y1 :0
 REFERINTA SIMPLA (0=NU;1=DA)?
 Y10:0
 REF.PE INEL (0=NU;1=DA)?
 Y10:0
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURA FRAZA
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CHEIA1-100
 Y2 :1
 CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
 Y10:1
 VALDARE ZECIMALA (0=NU;1=DA)?
 Y10:1
 VALDARE ZECIMALA 0.50
 SIR DE CARACTERE (0=NU;1=DA)?
 Y10:1
 CUVINT 1 CARACTERE
 Z3 :A
 REF.PE INEL (0=NU;1=DA)?
 Y10:1
 LEG CHEIA:B30701000 NR.5 VALDARE:0 CU INEL 0 ENT. REFERA SIMPLU
 REFERA PE INEL
 DORITI REFERIREA (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURA ENTITATE
 * CREARE (0=NU;1=DA) ?
 Y10:0
 * POZITIONARE/STERGERE
 SE INVESTIGHEAZA SUBANSAMBLUL 1
 CONTINE 12 ENTITATI
 ENTITATE CHEIA:COND.OPERATIYA A PRODUCIEI VALDARE:NU
 CONTINUATI (0=NU;1=DA) ?
 Y10:1
 SE STERGE IN INTREGIME (0=NU;1=DA)
 Y1 :0
 SE POZITIONEAZA (0=NU;1=DA)
 Y1 :1
 ENTITATE CHEIA:ESTE CAZ PARTICULAR AL VALDARE:PO
 CONTINUATI (0=NU;1=DA) ?
 Y10:0
 DENUMIRE IN STRUCTURA-
 SE COBOARA UN NIVEL IN STRUCTURA
 DENUMIRE IN STRUCTURA LEG
 * CREARE (0=NU;1=DA) ?
 Y10:



0
 Y10:1
 CHEIA:000101000-991231999
 Y2 :830701000
 EXISTA IN BAZA URMATOARELE ENTITATI CU ACEASTA CHEIE:
 LEG CHEIA:830701000 NR.5 VALOARE:0 CU INEL1 ENT. REFERA SIMPLU
 REFERA PE INEL
 CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
 Y10:1
 VALOARE INTREAGA (0=NU;1=DA)?
 Y10:0
 REFERINTA SIMPLA (0=NU;1=DA)?
 Y10:0
 RLF.PE INEL (0=NU;1=DA)?
 Y10:1
 FRAZA CHEIA:1 VALOARE:0.5 CU INEL0 ENT. REFERA PE INEL LEG AVIND
 CHEIA:830701000 NR.5 VALOARE:0 CU INEL1
 ENT.
 DORITI REFERIREA (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURAENTITATE
 * CREARE (0=NU;1=DA) ?
 Y10:0
 * POZITIONARE/STERGERE
 SE INVESTIGHEAZA SUBANSAMBLUL 1
 CONTINE 13 ENTITATI
 ENTITATE CHEIA:COND.OPERATIVA A PRODUCTIEI VALOARE:ND
 CONTINUATI (0=NU;1=DA) ?
 Y10:1
 SE STERGE IN INTREGIME (0=NU;1=DA)
 Y1 :0
 SE POZITIONEaza (0=NU;1=DA)
 Y1 :0
 ENTITATE CHEIA:ESTE CAZ PARTICULAR AL VALDARE:PO
 CONTINUATI (0=NU;1=DA) ?
 Y10:1
 SE STERGE IN INTREGIME (0=NU;1=DA)
 Y1 :0
 SE POZITIONEaza (0=NU;1=DA)
 Y1 :1
 ENTITATE CHEIA:PROCES DE CONDUCERE VALDARE:ND
 CONTINUATI (0=NU;1=DA) ?
 Y10:0
 DENUMIRE IN STRUCTURA
 SE COBOARA UN NIVEL IN STRUCTURA
 DENUMIRE IN STRUCTURALEG
 * CREARE (0=NU;1=DA) ?
 Y10:1
 CHEIA:000101000-991231999
 Y2 :830701000
 EXISTA IN BAZA URMATOARELE ENTITATI CU ACEASTA CHEIE:
 LEG CHEIA:830701000 NR.5 VALOARE:0 CU INEL1 ENT. REFERA SIMPLU
 REFERA PE INEL

LEG CHEIA:830701000 NR.6
 VALOARE:0 CU INEL0 ENT. REFERA SIMPLU
 REFERA PE INEL FRAZA AVIND CHEIA:1
 VALOARE:0.5 CU INEL1 ENT.

CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?

Y10:1

1

VALDARE INTREAGA (0=NU;1=DA)?

Y10:0

REFERINTA SIMPLA (0=NU;1=DA)?

Y10:1

ENTITATE CHEIA:COND.OPERATIVA A PRODUCTIEI VALDARE:ND

DORITI REFERIREA (0=NU;1=DA)?

Y10:1

REF.PE INEL (0=NU;1=DA)?

Y10:1

FRAZA CHEIA:1 VALDARE:10.5 CU INEL1 ENT. REFERA PE INEL LEG AVIND
CHEIA:830701000 NR.5 VALDARE:0 CU INEL1
ENT.

DORITI REFERIREA (0=NU;1=DA)?

Y10:1

* CREAT

DORITI MEMORAREA IN SUBANSAMBLUL 1 ?

Y10:1

* INREGISTRAT

DORITI MEMORAREA IN SUBANSAMBLUL 2 ?

Y10:1

* INREGISTRAT

DORITI MEMORAREA IN SUBANSAMBLUL 3 ?

Y10:1

* INREGISTRAT

DENUMIRE IN STRUCTURAENTITATE

* CREARE (0=NU;1=DA) ?

Y10:0

* POZITIONARE/STERGERE

SE INVESTIGHEAZA SUBANSAMBLUL 1

CONTINE 14 ENTITATI

ENTITATE CHEIA:COND.OPERATIVA A PRODUCTIEI VALDARE:ND

CONTINUATI (0=NU;1=DA) ?

Y10:1

SE STERGE IN INTREGIME (0=NU;1=DA)

Y1 :0

SE POZITIONEAZA (0=NU;1=DA)

Y1 :0

ENTITATE CHEIA:ESTE CAZ PARTICULAR AL VALDARE:PO

CONTINUATI (0=NU;1=DA) ?

Y10:1

SE STERGE IN INTREGIME (0=NU;1=DA)

Y1 :0

SE POZITIONEAZA (0=NU;1=DA)

Y1 :0

ENTITATE CHEIA:PROCES DE INDUCERE VALDARE:ND

CONTINUATI (0=NU;1=DA) ?

Y10:1

SE STERGE IN INTREGIME (0=NU;1=DA)

Y1 :0

SE POZITIONEAZA (0=NU;1=DA)

Y1 :1

DENUMIRE IN STRUCTURA-

SE COBOARA UN NIVEL IN STRUCTURA

DENUMIRE IN STRUCTURALEG

* CREARE (0=NU;1=DA) ?

Y10:1

CHEIA:800101000-991231999

Y2 :830701000

EXISTA IN BAZA URMATOARELE ENTITATI CU ACEASTA CHEIE:

LEG CHEIA:830701000 NR.5 VALDARE:0 CU INEL1 ENT. REFERA SIMPLU
REFERA PE INELLEG CHEIA:830701000 NR.6
VALDARE:1 CU INEL0 ENT. REFERA SIMPLU
REFERA PE INEL FRAZA AVIND CHEIA:1
VALDARE:10.5 CU INEL2 ENT.LEG CHEIA:
830701000 NR.2 VALDARE:1 CU INEL0 ENT.
REFERA SIMPLU ENTITATE AVIND CHEIA:
COND.OPERATIVA A PRODUCTIEI VALDARE:ND
REFERA PE TNFI FRAZA AVIND CHEIA:1

VALOARE:A0.5 CU INEL2 ENT.
 CHEIA DATA RAMINE VALABILA (0=NU;1=DA)?
 Y10:1
 VALOARE INTREAGA (0=NU;1=DA)?
 Y10:0
 REFERINTA SIMPLA (0=NU;1=DA)?
 Y10:1
 ENTITATE CHEIA:COND.OPERATIVA A PRODUCTIEI VALOARE:ND
 DORITI REFERIREA (0=NU;1=DA)?
 Y10:0
 ENTITATE CHEIA:ESTE CAZ PARTICULAR AL VALDARE:P0
 DORITI REFERIREA (0=NU;1=DA)?
 Y10:1
 REF.PE INEL (0=NU;1=DA)?
 Y10:1
 FRAZA CHEIA:1 VALOARE:A0.5 CU INEL2 ENT. REFERA PE INEL LEG AVIND
 CHEIA:B30701000 NR.5 VALOARE:0 CU INEL1
 ENT.
 DORITI REFERIREA (0=NU;1=DA)?
 Y10:1
 * CREAT
 DORITI MEMORAREA IN SUBANSAMBLUL 1 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 2 ?
 Y10:1
 * INREGISTRAT
 DORITI MEMORAREA IN SUBANSAMBLUL 3 ?
 Y10:1
 * INREGISTRAT
 DENUMIRE IN STRUCTURA*
 \$

19

GO
 QUESTION: SIGLEINTELEC 10
 * LSTINTELEC ?
 INTELEC 01/07/83
 TEXTULCOND.OPERATIVA A PRODUCTIEI
 DATA-VERSIUNEA (9 CIFRE)830701000
 MEMORARE INMMEA

COND.OPERATIVA A PRODUCTIEI ESTE CAZ PARTICULAR AL PROCES DE CONDUCERE .
 *LOCATIE INITIALA:1
 *LOCATIE FINALA :4
 TEXTUL*
 \$LOGO
 LOGOUT LE:01/07/83 A 19.36.48.94.
 SOCRATE-ITC VER.04/05/83

199

EXPLICĂȚII

1. Crearea matricei auxiliare de memorare MEMA a textelor de limbaj INTELEC introduse sau extrase din BDR.
2. Crearea elementului de structură denumit ENTITATE, în care se memorează cuvintul în limbaj INTELEC precum și caracteristicile sale :
 - sir de caractere, număr, variabilă;
 - tip : noțiune, predicat, calificator, operator (N,P,C,O).
3. Crearea elementului de structură denumit INDEX, în care se memorează cuvintele cheie de regăsire ale unui cuvânt de limbaj INTELEC.
4. Crearea elementului de structură denumit FRAZA, în care se memorează referința unei fraze la inelul de text de care aparține, în care se inscrie inelul cuvintelor ce formează fraza și care conține caracteristicile frazei :
 - cheie de regăsire (numericală);
 - gradul de generalitate (D,P,G,A);
 - probabilitatea de adevăr (intre 0 și 1, cu două zecimale).
5. Crearea elementului de structură denumit LEG, care asigură legătura cuvintului de limbaj INTELEC, fie ca o entitate text, fie ca o componentă a unei fraze și are următoarele caracteristici :
 - cheie numerică de regăsire a textului, formată din 9 cifre AALLZZVVV care reprezintă : AA = ultimile două cifre ale anului, LL = luna, ZZ = ziua, VVV = versiunea (0÷999);
 - număr întreg reprezentând ultima versiune a textului, existent în bază la data respectivă.
6. Crearea, în limbaj gazdă, a legăturii frazei la entitatea text.
7. Listarea structurii create în BDR.
8. Introducerea în BDR a entității de tip *noțiune* <COND. OPERATIVA A PRODUCTIEI> și memorarea ei în subansamblele de selecție 1, 2 și 3.
9. Introducerea în BDR a cuvintelor cheie de regăsire pentru entitatea de mai sus : CONDUCERE, CONDUCEREA PRODUCTIEI, COND. OPERATIVA A PRODUCTIEI, INTREPRINDERE.
10. Introducerea în BDR a entității de tip *predicat* [ESTE CAZ PARTICULAR AL] și memorarea ei în subansamblele de selecție 1, 2 și 3.
11. Introducerea în BDR a cuvintului cheie de regăsire PREDICAT, pentru entitatea de mai sus.
12. Introducerea în BDR a entității de tip *noțiune* <PROCES DE CONDUCERE> și memorarea ei în subansamblele de selecție 1, 2 și 3.
13. Introducerea în BDR a cuvintelor cheie de regăsire : PROCES SI CONDUCERE, pentru entitatea de mai sus.
14. Crearea legăturii pentru entitatea text <COND. OPERATIVA A PRODUCTEI>, avind cheia numerică : 830701000 (anul = 83, luna = 07, ziua = 01, versiunea = 000) și versiunea maximă existentă 0.
15. Crearea inelului pentru fraza <COND. OPERATIVA A PRODUCTEI> [ESTE CAZ PARTICULAR AL] <PROCES DE CONDUCERE>, care referă entitatea text <COND. OPERATIVA A PRODUCTEI> prin legătura mai sus creată.

16. Crearea legăturii pentru primul cuvint al frazei : <COND. OPERATIVA A PRODUCTIEI>. Se menține cheia numerică 830701000 a entității text.
17. Crearea legăturii pentru cel de-al doilea cuvint al frazei : [ESTE CAZ PARTICULAR AL]. Se menține cheia numerică acordată entității text 830701000. Se face referire la entitatea precedentă din frază.
18. Crearea legăturii pentru cel de-al treilea cuvint al frazei : <PROCES DE CONDUCERE>. Se menține aceeași cheie numerică 830701000 acordată anterior entității text ; Se face referire la entitatea precedentă din frază.
19. Terminarea înregistrării în BDE.
20. Căutarea textului <COND. OPERATIVA A PRODUCTIEI> din 1.07.1983, versiunea 0, în BDR și listarea frazei înregistrate la acest text.

BIBLIOGRAFIE

1. MARX, ENGELS, *Dialectica naturii*, Opere, vol. 20, Edit. politică, Bucureşti, 1964.
2. V. I. LENIN, *Caiete filozofice*, Opere complete, vol. 29, Edit. politică, Bucureşti, 1966.
3. ARISTOTEL, *Metafizica* (XIII, 1078 b, cap. 4) (trad. în lb. română de St. Bezdechi), Edit. Academiei, Bucureşti, 1965.
4. ARISTOTEL, *Despre susțință* (Cartea a III-a, 431 b), (trad. în lb. română și note de N. I. Ștefănescu), Edit. științifică, Bucureşti, 1969.
5. ARISTOTEL, *Organon I* (trad. în lb. română și studiu introductiv și note de Mircea Florian), Edit. științifică, Bucureşti, 1957.
6. ARISTOTEL, *Organon IV. Respingerile sofistice* (trad. în lb. română, studiu introductiv și note de Mircea Florian), Edit. științifică, Bucureşti, 1957.
7. ACKOFF, R. L., SASIENI, M.V., *Bazele cercetării operaționale*, Edit. tehnică, Bucureşti, 1975.
8. AHO, A.V., ULLMAN, J.D., *The Theory of Languages*, Mathematical Systems Theory, 2, Springer—Verlag, New-York Inc. (1968).
9. ARSENI, C., *Competență și performanță informațională în organizarea funcțională a creierului uman*, Comunicare la primul Simpozion Național de inteligență artificială și robotică, Bucureşti, 22—23 oct., 1981.
10. BACON, FRANCIS, *Noul Organon* (Cartea I, LXX), (trad. în lb. română), Edit. Academiei, 1957.
11. BOLDUR, GH., PETRESCU N. și FLORIAN M., *Fundamentarea complexă a procesului decizional economic*, Edit. științifică, Bucureşti, 1976.
12. BOLDUR, GH., BĂNCILĂ, I., *Metode și mijloace moderne de luare a deciziilor în întreprinderi* (sinteză documentară), I.D.T., Bucureşti, 1974.
13. BĂLĂCEANU, C. și NICOLAU, EDM., *Les fondements cybernétiques de l'activité nerveuse*, Expansion Scientifique Française, 1971.
14. BUFFA, E.S., *Conducerea modernă a producției* (vol. II), Edit. tehnică, Bucureşti, 1975.
15. BIEGEL, J. E., *Previziunea, planificarea, programarea și controlul activităților industriale*, în *Manual de inginerie industrială*, editor șef H. B. Maynard (vol. II), Edit. tehnică, Bucureşti, 1976.
16. BOBROW, D., COLLINS, A., *Representation and understanding*, New York, Academic Press, 1975.
17. CONSTANTINESCU, P., *Clasificarea unei mulțimi de elemente în raport cu o mulțime de proprietăți*, Revista de studii și cercetări de calcul economic și cibernetică economică, 3 (1967).
18. CONSTANTINESCU, P., *Sistemul informatic model al întreprinderii industriale*, Academia Ștefan Gheorghiu, Bucureşti, 1977.
19. CONSTANTINESCU, P., GUTU, ST., *Modele și sisteme procedurale în conducerea întreprinderilor*, Revista AMC, 34 (1981).
20. CORDUCK, P. MC., *Machines Who Think*, W. H. Freeman and Co, San Francisco, 1979.
21. CONSTANTIN, D., *Inteligența materiei*, Edit. militară, Bucureşti, 1981.
22. CHARMIAK, E., WILKS, Y., *Computational semantic, an introduction to artificial intelligence and natural language comprehension*, North Holland Publishing, Amsterdam, 1976.
23. CYERT, R. M., MARCH, J. G., *A behavioral theory of the Firm*, Prentice Hall Englewood Cliffs, New Jersey, 1963.
24. DRĂGĂNESCU, M., *A două revoluție industrială. Microelectronica, automatica, informatica — factori determinanți*, Edit. tehnică, Bucureşti, 1980.

25. DRĂGĂNESCU, M., *Sistem și civilizație*, Edit. politică, București, 1976.
26. DRĂGĂNESCU, M., *Profunzimile lumii materiale*, București, Edit. politică, 1979.
27. DUMITRIU, ANTON, *Teoria logicii*, Edit. Academiei, 1973.
28. DUMITRIU, ANTON, *Istoria logicii*, Edit. didactică și pedagogică, București, 1975.
29. DUMITRĂȘCU, L., IOACHIM, AL., *Tehnici de construire a programelor cu structuri alternative*, Edit. Academiei, București 1981.
30. DUMITRESCU, M., OPREA, N., PLEȘOIANU, G., *Proiectarea sistemelor de organizare și conducere*, Edit. științifică și enciclopedică, București, 1978.
31. FILMORE, GH., *Types of lexical information, Studies in Syntax and Semantics*, Ed. by V. Kiefer, Dordrecht, Holland, 1969.
32. FELDMAN, KANTERS, *Organisational decision making*, Handbook of Organization, Chicago, Rand Mc. Nally Co., 1965.
33. FORRESTER, J. W., *Industrial Dynamics*, MTT Press, Cambridge – Massachusetts, 1961.
34. FRANCIS, M. F., *Computer Modeling and Simulation*, Prentice-Hall, Inc. 1968.
35. GUȚU, ST., Programarea producției cu ajutorul sistemului „SOETPLAN” la unele întreprinderi constructoare de mașini, Revista AMC 35 (1983) pag. 162–173
36. GUȚU, ST., TAŞNADI, AL., PĂUN, GH., STAN, E., *Metode de optimizare a deciziilor în activitatea economico-socială. Aplicații pe terminal*, Acad. Ștefan Gheorghiu, 1983.
37. GUȚU, ST., *Contribuții la elaborarea sistemului informatic al conducerii operațive a producției* (cu aplicații în industria constructoare de mașini), Teză de doctorat, I. P. București, 1980.
38. GUȚU, ST., *Generator de funcții pentru o bază de date*, Studii și cercetări de calcul economic și cibernetic economică, 4 (1978).
39. GUȚU, ST., *La Science administrative et l'informatique. Un modèle de système informatique pour la recherche administrative*, Thèse de doctorat de spécialité (3-ème cycle), Université Paris I, Sorbonne, 1974.
40. GEOFFREY, G., *System Simulation*, Prentice-Hall, 1969.
41. GEORGESCU, I., *Structura cognitivă și rezolutivă a sistemelor inteligeante de tip expert*, Comunicare la primul Simpozion național de inteligență artificială și robotică, București, 22–23 oct. 1981.
42. GEORGESCU, I., SITARU, C., *Sisteme de proiectare asistată de calculator, bazate pe reprezentarea cunoașterii*, Buletinul Român de Informatică, 1 (1981).
43. GEORGESCU, I., *An approach to the fundamental features of non-von Neumann computers*, Rundboef Kunstliche Intelligenz, 23, Hamburg, April (1981).
44. HOMOȘ, T., *Optimizarea principaliilor parametri ai programării operațive a producției*, Construcția de mașini, 23, 11 (1971).
45. HOMOȘ, T., CIOBANU, GH., *Stabilirea sistemului de producție în construcția de mașini*, Revista de statistică, 1 (1971).
46. HOMOȘ, T., *Optimizarea principaliilor parametri ai programării operațive a producției*, Construcția de mașini, 23, 11 (1971).
47. HOMOȘ, T., *Legile și principiile organizării proceselor de producție în spațiu și timp*, Institutul politehnic din București, Catedra de organizare și conducere industrială, 1970.
48. HOMOȘ, T., *Principiile de bază de organizare în timp și spațiu a producției în construcțiile de mașini*, Construcția de mașini, 12 (1970).
49. HUANG, SHENG-CHAO, GOEL, A. L., *An analytical model for information processing systems*, National Computer Conference, U.S.A., 1974 – Research supported by RADC – Rome, Contract Numb. F – 30602 – 72 – C – 0281.
50. HUNT, E. B., *Artificial Intelligence*, New York, Academic Press, 1975.
51. HUTTENLOCHER, I., *Constructing spatial images: a strategy in reasoning*, in *Thinking-Readings in Cognitive Science*, 1977.
52. JOHNSON-LAIRD, P. N., WASON, P. C., *Thinking-Readings in Cognitive Science*, Cambridge University Press, Cambridge, London, New York, Melbourne, 1977. v.I
53. JOHNSON-LAIRD, P. N., WASON, P. C., *An introduction to the scientific study of thinking*, in *Thinking-Readings in Cognitive Science*, Cambridge University Press, London, New York, Melbourne, 1977, vol. I.
54. JOHNSON-LAIRD P. N. and WASON, P. C. (editors), *Thinking-Readings in Cognitive Science*, Cambridge University Press-Cambridge, New York, Melbourne, 1977, vol. II.
55. KAMI, M. J., *Planificarea și controlul prin metoda sistemelor*, în *Probleme actuale ale conducerii întreprinderilor*, Edit. politică, București, 1973.
56. KAUFMANN, A., *Metode și modele ale cercetării operaționale* (vol. I), Edit. științifică, București, 1965.

57. KANT, IMMANUEL, *Critica rațiunii pure* (trad. în lb. română de Nicolae Bagdasar și Elena Moisuc). Edit. științifică, București, 1969.

58. KAUFMANN, A., *Metode și modele ale cercetării operaționale* (vol. II), Edit. științifică, București, 1967.

59. LADRIÈRE, J., *Les limitations internes des formalismes*, Paris, Louvain, 1957.

60. MARCH, J. G., SIMON, H. A., *Les organisations*, Dunod, Paris, 1969.

61. MOON, D. A., *MACLISP Reference Manual*, Project MAC Massachusetts Institute of Technology, Cambridge, Massachusetts, 1974.

62. MINSKY, M., *Frame – System theory*, Thinking-Readings in *Cognitive Science*, 1971.

63. NICOLAU, EDM., BĂLĂCEANU, C., *Elemente de neurocibernetică*, Edit. științifică, București, 1967.

64. NEWEL, A., SIMON, H. A., *Human Problem Solving*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1972.

65. NEMETI, L., *Metode euristicice deordonanțare a fabricașiei*, Edit. Dacia, Cluj-Napoca, 1977.

66. POSPELOV, D. A., *Abordarea sistematică a modelării activității intelectuale în Metoda cercetării sistemice*, Edit. științifică, București, 1974.

67. POSPELOV, D. A., *Artificial Intelligence Systems: Results and short term Outlook*, Problems of Control and Information Theory, 9, 1 (1978).

68. PIAGET, J., *Psihologia inteligenței*, Edit. științifică, București, 1965.

69. ROMAN, V., DRĂGĂNESCU, M., *Evoluția inteligenței artificiale, a inteligenței umane și a corelației dintre ele. Consecințele sociale ale tehnologiei inteligenței artificiale*, în *Știința și contemporaneitatea*, Edit. politică, București, 1980.

70. SFEZ, LUCIAN, *Critique de la decision*, Cahiers de la fondation nationale des sciences politiques, Armand Colin, Paris, 1970.

71. TREITELMAN, W., *INTERLISP Reference Manual*, Xerox Palo Alto, Research Center, Palo Alto, California sau Beraneck and Newman, Cambridge, Massachusetts 1974.

72. WINSTON, P. H., BROWN RICHARD, H. (editors), *Artificial Intelligence: An MIT Perspective* (vol. I), The MIT Press, Cambridge, Massachusetts and London, England, 1979.

73. WINSTON, P. H., *Artificial Intelligence*, Addison Wesley Publishing Company Inc., 1977.

74. WINSTON, P. H., HORN, B.K.P., *LISP*, Addison, Wesley Publishing Co, 1981.

75. WINSTON, P. H. (editor), *The psychology of computer vision*, New York, Mc. Graw-Hill, 1975.

76. WEIZENBAUM, J., *Computer power and human reason. From judgement to calculation*, San Francisco W. H. Freeman and Co., 1976.

77. WINOGRAD, T., *Formalisms for knowledge*, Thinking-Readings in *Cognitive Science*, 1971.

78. WINOGRAD, T., *Understanding natural language*, N. Y. Academic Press, 1972.

79. XENOFON, *Memorabilele* (IV, 6, 15) în *Istoria logicii* (op. citată).

80. * * * *Grand Larousse encyclopédique*, Librairie Larousse, Paris, 1962, tome sixième.

81. * * * *SOCRATE – Manuel d'opération*, Juillet, 1973.

82. * * * *SOCRATE – Manuel d'utilisation*, 4336 E/FR, Juillet, 1973.

83. * * * *Système de chainage des Fichiers (S.C.F.) manuel d'utilisation et d'opération*, 4025 E 1/FR, Janvier, 1972.

INDICE DE TERMENI

A

abaterea obiectivului realizat, 27
act decizional, 26, 27, 28, 29, 30
actul de autoreflexie al găndirii, 12
accidentul, 14, 15
adoptarea soluției, 27, 28
ajustarea manuală finală, 82, 85, 86
alegere satisfăcătoare, 35
algebra logică, 18
algoritm, 23, 70, 71, 72, 77, 92
 ● de programare convergent, 98
 ● euristic de simulare a producției, 99
alternative, 22, 23, 26, 27, 28, 29, 30, 35
analizor
 ● sintactic, 179, 181
 ● semantic, 153, 180
antiteza, 17
aprecieri ale rezultatelor, 89, 96
arhitectura sistemului, 128
asertiune, 43
armonizare globală, 97
assimilis, 180
attracție, 17
automat programabil, 78
autoritate, 14
axiome, 19

B

baza de date relațională (B.D.R.), 37, 35, 42, 57, 58, 59, 60, 62, 66, 67, 68, 73, 78, 80, 81, 87, 90, 163, 167, 180
BENSON, 131
bilanțuri capacitate-încărcare, 131
BOMP, 98

C

cadru relațional, 37, 68, 69, 70, 153, 162
calculatorul electronic, 16, 25, 35, 78, 97, 98, 99, 100, 105, 128, 153, 154
 ● consilier, 95, 101

 ● executant, 95, 101, 164
 ● robot, 95, 101, 153, 165
 ● de proces, 28, 166
calculul manoperei, 147
 ● comenziilor de reparații, 148
 ● parametrilor desfășurării producției, 73, 74, 75, 76, 96, 99
calificator, 42, 44, 45, 48, 49, 50, 54, 58, 62, 64, 65, 68, 81
calitate, 13, 15, 17
capacitatea medie a posturilor, 144
categorie, 13, 14, 15, 17, 18, 22, 23
cauză, 15, 18, 23
celule nervoase, 20, 21
CENTRONICS, 187
ciclogramă
 ● de reparații, 147
 ● de fabricație, 83, 86, 140
ciclu de fabricație, 83, 97, 117
clasa de decizie, 26
CLASS, 98, 99, 114, 159
cod, 65, 66, 67, 68
codificare, 145
coeficientul sistemului de producție, 107
colaborări, 98
comandă, 44, 45, 49, 51, 55, 60, 65, 66, 67, 68, 72, 82, 83, 84, 86, 87, 89, 91, 92, 97, 100, 153
 ● condiționată, 44, 45, 50, 55
 ● internă, 45, 48, 80, 97
 ● de realizare a sarcinilor de producție, 96
 ● produs, 118
complement, 42, 46, 47, 48, 49, 50, 51, 52, 53, 54, 68, 69, 70, 81, 82, 180
concatenare, 38
conceptul pur, 15
 ● de raționalitate a deciziei, 35
 ● de multiraționalitate, 25, 35, 36
condiții, 28, 29, 30
 ● restricții, 60, 64, 79, 80, 81, 82, 83, 85, 86, 87, 89, 90, 91, 93, 94, 96
 ● generale de utilizare, 114
conducerea operativă a producției, 22, 23, 31, 36, 37, 40, 42, 44, 58, 73, 74, 75, 77, 78, 82, 95, 96, 99, 100, 101, 105, 108, 113, 114, 153, 154
conduită, 20

INTELEC, 37, 42, 46, 58, 68, 80, 90, 92, 113, 153, 164, 187
 inteligență, 16, 23, 78
 ● artificială, 22, 100
 ● economică, 14
 ● superioară, 12
 ● umană, 12, 14, 16
 intenții, 14
 intentio
 ● prima, 14
 ● secundo, 14
 interactiv, 14
 intuiție, 15, 16, 18, 100
 invariant, 14
 ipoteza, 14, 16, 79
 ironia, 12
 iterativ
 ● reușită, 84
 ● nereușită, 84

I

îmbunătățire a operației de selecție, 92
 încărcarea.
 ● de bază, 110
 ● comenzi-produs, 117
 ● inițială a posturilor de lucru, 97, 122
 întâmplare, 18

J

judecată, 14, 15, 19, 77

L

lansarea documentației, 143
 lanțul operațiilor tehnologice, 142
 legătură directă, 73
 ● de tip R-I, 73
 ● de ordinul 1, 73, 74
 ● de ordinul 2, 74
 LISP, 153, 181, 183
 limbaj intermediar, 168
 limbaje specializate, 100
 limbaj procedural complet al instrumentului informatic, 167
 lista
 ● reperelor (pieselor), 98
 ● subansamblelor, 98
 ● ansamblelor, 98
 ● produselor, 98

logica
 ● modernă, 19
 ● formală, 18, 19

luarea deciziei, 27

lungimea ciclului de fabricație al produsului, 99, 117

M

maieutica, 12
 manoperă pe post de lucru, 96
 MATCH, 182
 materie, 12, 14, 17, 18
 materii prime, 98
 materiale, 98
 matricea
 ● booleană, 34
 ● consecințelor, 26
 media variabilei aleatoare, 125
 mediu decizional, 28
 memorii
 ● dinamice, 16
 ● virtuale, 167
 metaconector, 45
 metalimbaj, 45, 168
 metoda
 ● narrativă, 30
 ● organigramei, 30
 ● tabeli de decizie, 30
 ● ELECTRE, 25, 33, 34, 35, 94
 ● aplicată pe terminal, 31
 ● indicilor de constanță a fabricației, 105
 ● deordonanțare
 manuală, 99
 automată, 99
 ● directă, 99
 model, 23, 25
 ● matematic, 25, 115
 ● semiotic, 15, 16
 ● de decizie, 28, 95
 ● INTELEC, 37, 46, 77, 78, 89, 93, 94, 95, 111, 153, 167
 ● relațional, 13, 37
 ● probabilistic, 13
 ● intelligent, 22, 23, 95, 100, 113, 153, 165
 module de optimizare, 128
 ● care translatează, 128
 ● de creare a bazei de date, 131
 ● de completare a bazei de date, 131
 ● de actualizare a bazei de date, 131
 ● de listare a încărcărilor de manoperă, 131

N

necesitate, 15, 18
 neuroni, 20
 noțiune internă, 42, 44, 45, 46, 47, 52, 53, 54, 58, 59, 64, 66, 69, 73

O

obiectiv
 ● realizat, 27
 ● planificat, 27



- de interferență, 108
- transcendentală, 15
- scene, 15
- scădere medie probabilistice, 120
- scop secundar, 35
- semifabricate, 98
- silogism, 14
- simulări, 89
- sintaxă, 18
- singular, 18
- sinteză, 17, 19
- sistem
 - informatic, 31
 - formal, 18, 19
 - de producție, 98, 105
 - procedural, 114
 - semiotic pseudofizic, 16
 - de semne de gradul
 - intii, 16
 - al doilea, 16
 - al treilea, 16
 - al patrulea, 16
 - intelligent de tip expert, 22
- situație
 - inedită, 89
 - dinainte prevăzută, 78
 - de progres, 76
 - inițială, 76
 - formală bine definită, 76
- SOCRATE, 153, 167
- soluție realizabilă, 73
- SOFTPLAN, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 77, 82, 98, 99, 103, 111, 113, 114, 153, 160
- sortimente, 96, 97
- „spargerea” unor comenzi, 128
- specia, 13
- structura capacităților disponibile, 57
- structură
 - cu un număr aproximativ egal de încărăcări de diferite mărimi, 166
 - cu un maxim de suprasarcină pronunțat și goluri foarte puține, 167
 - generalizată, 168
- studiu statistic al cazurilor reale, 39
- subiect, 20, 42, 45, 46, 47, 48, 50, 52, 53, 54, 66, 68, 70, 82
- subprogram, 48, 49, 50, 72, 77, 79, 80, 81, 82, 87, 88, 90, 91
- SUPERIOADA de programare, 83
- suportul informatic al limbajului, 42, 55
- supraîncărcare, 82, 84, 97, 98

T

- tabele de decizie, 25, 30, 31, 153, 180, 181
- tehnologie de grup, 64
- teoreme, 19, 46
- teoria
 - deciziei, 26, 27, 29, 78
 - utilității, 27, 35
 - statistică a deciziei, 27
 - deciziilor multidimensionale, 27
 - deciziilor de grup, 27
 - grafurilor, 33, 117
- termenul de livrare, 98
- terminal, 34, 37, 51, 58, 60, 87
- teză, 17
- timbul, 13, 15
 - de pregătire-Incheiere, 116
 - de transport, 116
 - de început cel mai devreme posibil al fabricației, 70, 117
 - de sfîrșit cel mai tîrziu admis pentru fabricație, 70, 117
- tipuri de produse, 96
- translație, 118, 127

U

- urmărirea realizărilor, 76
- utilizarea prin INDEX, 66
- utilizarea optimă a unui pachet de programe, 78

V

- valoare semantică unică, 38
- valoarea
 - medici probabilistice, 124
 - priorității comenzi, 98
 - producției neterminate medii ciclice, 99
- variante
 - de program realizabile, 97
 - de structură a planului de producție, 97
 - de încărcare, 147
- volumul
 - anual de piese, 106
 - comenziilor de reparații, 147

STELIAN GUTU, LIVIU DUMITRASCU. Elemente de inteligență artificială pentru conduceerea operativă a producției (Artificial intelligence elements used in industrial production management), BUCUREȘTI, EDITURA ACADEMIEI REPUBLICII SOCIALISTE ROMÂNIA, 1983, 220 p.

ABSTRACT

Notwithstanding the fact that the intelligence is an essential characteristic of the human nature and a paramount factor of the evolution of life on earth, it continues to be for the most part unknown, even nowadays.

The human being disposes at present of highly improved tools which are able to work independently with good accuracy accordingly to the intelligent programs created by men. There exist at present electronic computers capable of taking over some functions of the human thought, based on memory and calculation.

Whether these computers are, indeed, intelligent? It is difficult to answer. At present there exist, no doubt, hosts of researchers who investigate an eventual extension of intelligence beyond the human limits, so that some objects and mechanisms may be able to carry out intelligent actions.

This book gives several ideas and original results in the field, with special reference to the industrial production management problem.

Starting from necessity to express the thought — therefore to re-evaluate intelligence — in language formulation, the authors propose an instrument for representation and handling of economical knowledge. This instrument, named INTELEC, is used throughout this book. INTELEC uses simplified language — in comparison with the natural language — realised on electronic computer. The informational support of the INTELEC model is a relational data base, whose flexible structure allows the acquisition, updating and erasing of the characteristics without affecting the existing information.

The use of the data base with the aid of the INTELEC language is illustrated by several examples. In filling the detailed information we proceed from more general sentence which is used as a pattern. It is also possible to synthesize some detailed informations and hence to obtain further information. Data retrieval is either direct (by combining the characteristic key words), or resorts to indirect methods, such as research

based on predicates : [IS PARTICULAR CASE OF], [IS (IN THIS CASE)], [IS MADE UP OF (ASSAMBLEY)], [IS MADE UP OF (SEQUENCE)] and also the processing of relational frames (see chapter 3).

The INTELEC model and its language are a practical tool which may be accommodated to the requirements of production management in enterprises. Based on the management concept developed in some universities and research centers in Romania, which recommends a thorough use of informatics in the planning-programming-launching-checking of the production activities — the authors of this book propose an original way of incorporating the computer in the realisation of this desideratum.

Thus, let us consider a computer which performs the management of the production in a certain enterprise. This computer may be defined as : (a) an adviser for all production management problems ; (b) an accomplisher of the production optimization programs, and (c) a computer-robot, obtained by combining the above two properties, to which a complex interdependent action between the computer and the environment is added.

The model which can develop through the above mentioned steps, from simple to more complex, taking over gradually some of the functions of the thought and human activities in the production management field, was defined as the intelligent model of operative management of production.

Mention should be made that man's role in the management and control of the computer work does not vanish. However, the importance of human actions diminishes with the increasing intelligent activities of the computer (see chapter 5).

Based on the intelligent model concept and using the INTELEC instrument, we have approached a concrete management case in the field of machinery building. For the small serial production and also for the production of the uniques objects we have used a package — made and experimented by one of the authors (S. Guțu) in some Romanian enterprises. This package, which turned out to be highly efficient in the planning-programming field, was applied at a global level (annually, quarterly) and on shorter time-spans (months, decades, weeks) for the type of production based on working machine groups. Using a relatively modest equipment, the above mentioned package enabled the manager to know in each instant with sufficient accuracy what product is really feasible from the entire production plan. The SOFTPLAN system (which is described in chapter 6) may be considered to be the main verification of the possibilities to perform the different plans (supply of raw materials, collaborations, delivery, etc.)

A complete model which gives answers to most of the questions about production management in machinery building has been transposed in the INTELEC language. In this manner the computer acting as an adviser can answer at questions partaking of the most important tasks to be solved. For instance, assuming that we want work with the SOFTPLAN package, the computer can furnish the detailed information about the procedure to follow, the program module to be selected for the next step, etc. This class of products is conservatively assigned to the artificial intelligence systems of "expert" type. The "force-intelligence" couple is realized, in this case, on the one hand by a better loading of the production capacity through SOFTPLAN, and by the generalized action model

⟨MAIN PROGRAM⟩ — ⟨SUBPROGRAM⟩ — ⟨SAVING⟩ — ⟨DECISION⟩ — ⟨LEARNING⟩ (see chapter 4) on the other. In fact, the two elements : “force” (the package) and “intelligence” (the intelligent model answering the user’s questions, work independently and are connected by the user, who informs the model of the results obtained with the package and awaits the computer’s instructions. The automatic connection between “force” and “intelligence” elements, possible in the future, could transform the model into a complex automation of robot type, capable to choose the variants suited to the aims of the entreprise and to follow their attainment.

The model described in this book is not only a theoretical one. Precise indications are given concerning the possibility to implement the INTELEC model and its relational data base (see chapter 7).

The creation and manipulation of the structure of the INTELEC model by means of a functional language carried out by authors and named Generalized Data Base Structure System (GDBSS) proves the viability of the suggested solution. As concerns the “force” share, the SOFTPLAN package has been experimented in several enterprises, the results being partially mentioned in this work (see chapter 6).

Finally, several recommendations pertain to the realization of the semantic and syntactic analyzers of INTELEC language, including an eventual use of the LISP language for which several programs are given.



Roughly speaking, this book is a collection of original ideas and solutions :

Following an introduction to intelligence in general and with reference to the artificial intelligence in particular, Chapter 2 gives definitions and axiomes relating to decision process.

The Chapter 3 provides a mathematical description of the INTELEC model (definitions, construction of the relational data base and some applications).

Chapter 4 approaches a special problem of artificial intelligence: behaviour planning with applications to the operative management of production. One defines the mechanism on which the applied part of the work is based — the general model of operative management of production. Next, the authors define the operative management concept and the SOFTPLAN package.

Finally, the connection between all these elements is discussed in the frame of the model of production management in the machinery building branch.

Although this book deals essentially with application, further uses of the INTELEC model and language can be envisaged in the artificial intelligence field. In this respect, the authors trust that an eventual development of the ideas given in this book would render this volume more consistent. As such the book appeals to a wide readership, including students, research workers, teachers, managers, engineers, since all information about “what” and “how” to manage production with the aid of artificial intelligence are rendered in an highly accessible language.

Additionally, the ideas expressed in this book may become the source of further valuable studies.

C O N T E N T S

CHAPTER 1. Characteristics of the human intelligence in the frame of economic activities of production

1.1. Intelligence and the thinking mechanism	12
1.2. The opportunity of introducing some artificial intelligence components into production management	22

CHAPTER 2. Models of decision processing

2.1. Preliminary considerations	26
2.2. About some mathematical formalizations in the decision problem	26
2.3. The dynamics of the decision process	27
2.4. Decisions having a routine character	28
2.5. Programming of the choice of alternatives	30
2.6. The utility concept; ELECTRE method used on terminals	31
2.7. The multi-rationality concept	35

CHAPTER 3. Knowledge representation and handling of data, using the INTELEC model: definition, structure, realisation, concept and use

3.1. Generalities	38
3.2. Definition of the INTELEC language	42
3.3. About predicates, predicative functions and theorems	46
3.4. Definition of relational data base	55
3.5. Principles of utilisation of the relational data base	57
3.6. Alternatives concerning the relational data base utilisations	58

CHAPTER 4. Behaviour planning in the frame of the INTELEC model

4.1. Generalities	78
4.2. Behaviour planning in the case of optimal utilisation of a package for operative management of production	78
4.3. Application in the case of utilisation of SOFTPLAN package	82
4.4. The introduction of some artificial intelligence elements in planning the decisions concerning the choice of the optimization stages	89

CHAPTER 5. Concept of intelligence applied in the production management problem

5.1. The structure of the operative management of production	96
--	----



5.2. The intelligent model concept	100
5.3. Use of the intelligent model concept in the operative management of the production field	102
CHAPTER 6. SOFTPLAN system: a basic component of an intelligent model of management in the machinery building branch	
6.1. System utility	114
6.2. Mathematical model	115
6.3. System architecture	128
6.4. Some results and conclusion obtained by the application of SOFTPLAN in enterprises	131
CHAPTER 7. Production management based on the INTELEC model	
7.1. Objectives	154
7.2. Computer in the adviser role	154
7.3. Computer in the accomplisher role	164
7.4. Computer-robot	165
7.5. INTELEC model realisation	167
ANNEXE	187
Bibliography	203
Subject index	207

Приложение 1. Структура и содержание книги

Литература, опубликованная в научно-технических журналах, а также в сборниках научных конференций

Литература, опубликованная в научно-технических журналах, а также в сборниках научных конференций

Redactor: EUGEN MARINEANU
Tehnoredactor: FELICIA BOLOCAN

*Bun de tipar: 1.XI.1983. Format: 16/70 × 100. colii de
tipar:*

*C.Z. pentru biblioteci mari: { 658: [153:007]
153: 007:658*

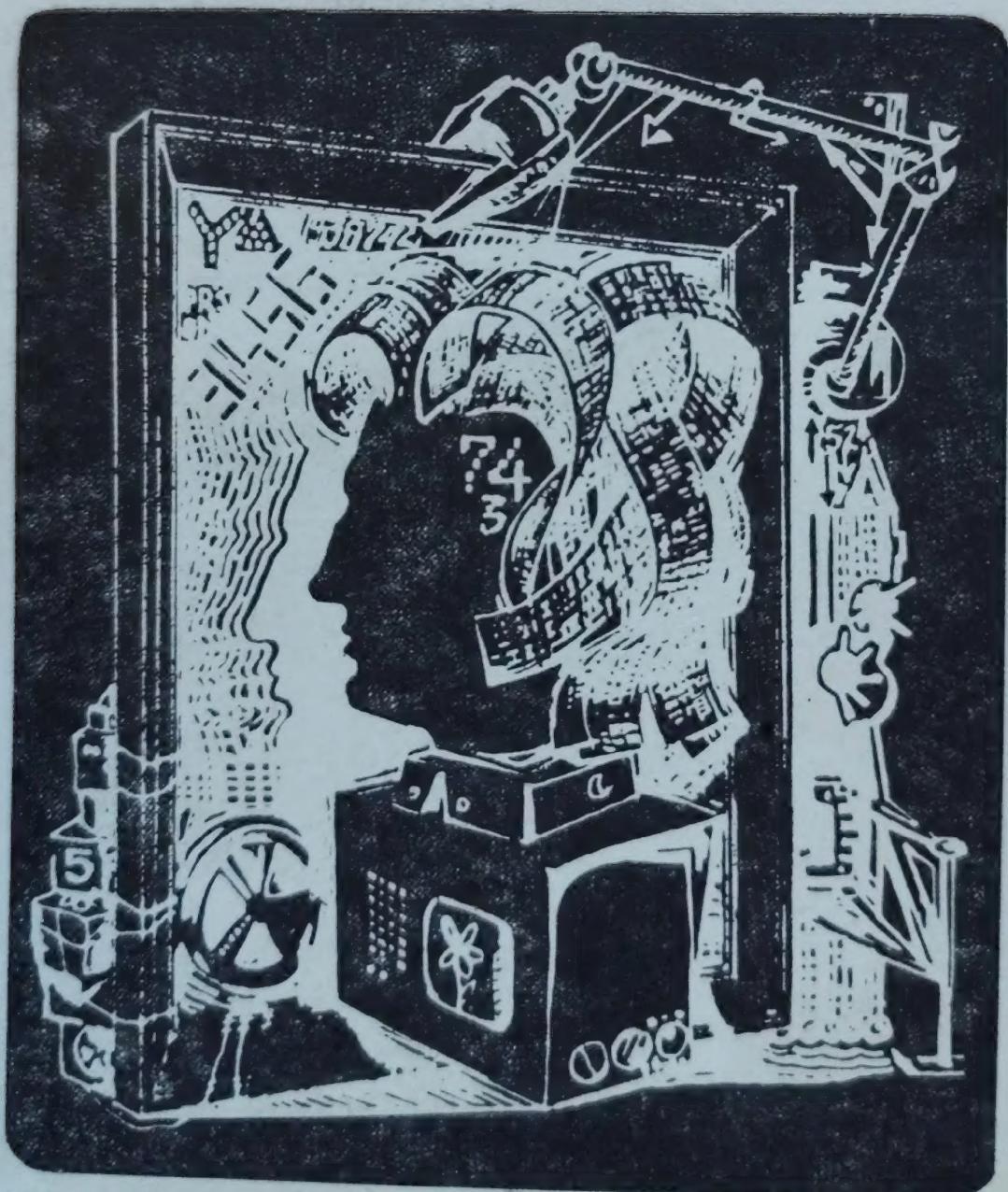
C.Z. pentru biblioteci mici: [681.142.83: 153]: 658



c. 219 I. P. Informația, Str. Brezoianu
nr. 23-25, București

EDITURA ACADEMIEI REPUBLICII SOCIALISTE ROMÂNIA

20



Lei 18,50



Scanned with OKEN Scanner